CR10 MEASUREMENT AND CONTROL MODULE
OPERATOR'S MANUAL

REVISION: 3/96

COPYRIGHT (c) 1987-1996 CAMPBELL SCIENTIFIC, INC.

WARRANTY AND ASSISTANCE

The **CR10 MEASUREMENT AND CONTROL MODULE** is warranted by CAMPBELL SCIENTIFIC, INC. to be free from defects in materials and workmanship under normal use and service for thirty-six (36) months from date of shipment unless specified otherwise. Batteries have no warranty. CAMPBELL SCIENTIFIC, INC.'s obligation under this warranty is limited to repairing or replacing (at CAMPBELL SCIENTIFIC, INC.'s option) defective products. The customer shall assume all costs of removing, reinstalling, and shipping defective products to CAMPBELL SCIENTIFIC, INC. CAMPBELL SCIENTIFIC, INC. will return such products by surface carrier prepaid. This warranty shall not apply to any CAMPBELL SCIENTIFIC, INC. products which have been subjected to modification, misuse, neglect, accidents of nature, or shipping damage. This warranty is in lieu of all other warranties, expressed or implied, including warranties of merchantability or fitness for a particular purpose. CAMPBELL SCIENTIFIC, INC. is not liable for special, indirect, incidental, or consequential damages.

Products may not be returned without prior authorization. To obtain a Returned Materials Authorization (RMA), contact CAMPBELL SCIENTIFIC, INC., phone (435) 753-2342. After an applications engineer determines the nature of the problem, an RMA number will be issued. Please write this number clearly on the outside of the shipping container. CAMPBELL SCIENTIFIC's shipping address is:

CAMPBELL SCIENTIFIC, INC.
RMA#_____
815 West 1800 North
Logan, Utah 84321-1784

CAMPBELL SCIENTIFIC, INC. does not accept collect calls.

Non-warranty products returned for repair should be accompanied by a purchase order to cover the repair.

CR10 MEASUREMENT AND CONTROL MODULE
TABLE OF CONTENTS

PAGE

OV1. PHYSICAL DESCRIPTION

OV1.1 Wiring Panel OV-1
OV1.2 Connecting Power to the CR10 ..... OV-5

OV2. MEMORY AND PROGRAMMING CONCEPTS

OV2.1 Internal Memory OV-5
OV2.2 CR10 Instruction Types OV-7
OV2.3 Program Tables, Execution Interval and Output Intervals OV-7

OV3. COMMUNICATING WITH CR10

OV3.1 CR10 Keyboard/Display OV-9
OV3.2 Using the PC208 Terminal Emulator (GraphTerm) OV-9
OV3.3 ASCII Terminal or Computer with Terminal Emulator OV-9

OV4. PROGRAMMING THE CR10

OV4.1 Functional Modes OV-10
OV4.2 Key Definition OV-10
OV4.3 Programming Sequence OV-11
OV4.4 Instruction Format OV-11
OV4.5 Entering a Program OV-12

OV5. PROGRAMMING EXAMPLES

OV5.1 Sample Program 1 ..... OV-13
OV5.2 Sample Program 2 ..... OV-14
OV5.3 Editing an Existing Program OV-15

OV6. DATA RETRIEVAL OPTIONS OV-17

OV7. SPECIFICATIONS OV-20

PROGRAMMING

1. FUNCTIONAL MODES

1.1 Program Tables - *1, *2, and *3 Modes 1-1
1.2 Setting and Displaying the Clock - *5 Mode 1-2
1.3 Displaying/Altering Input Memory, Flags, and Ports - *6 Mode 1-3
1.4 Compiling and Logging Data - *0 Mode 1-4
1.5 Memory Allocation - *A 1-4
1.6 Memory Testing and System Status - *B 1-6
1.7 *C Mode -- Security 1-7
1.8 *D Mode -- Save or Load Program 1-7

2. INTERNAL DATA STORAGE

2.1 Final Storage Areas, Output Arrays, and Memory Pointers 2-1
2.2 Data Output Format and Range Limits 2-3
2.3 Displaying Stored Data on Keyboard/Display - *7 Mode 2-3

3. INSTRUCTION SET BASICS

3.1 Parameter Data Types 3-1
3.2 Repetitions 3-1
3.3 Entering Negative Numbers 3-1
3.4 Indexing Input Locations and Ports 3-1
3.5 Voltage Range and Overrange Detection 3-2
3.6 Output Processing 3-2
3.7 Use of Flags: Output and Program Control 3-3
3.8 Program Control Logical Constructions 3-4
3.9 Instruction Memory and Execution Time 3-5
3.10 Error Codes 3-8

DATA RETRIEVAL/COMMUNICATION

4. EXTERNAL STORAGE PERIPHERALS

4.1 On-Line Data Transfer - Instruction 96 4-1
4.2 Manually Initiated Data Output - *8 Mode 4-3
4.3 Cassette Tape Option 4-3
4.4 Printer Output Formats 4-5
4.5 Storage Module (SM192/716) 4-6
4.6 *9 Mode -- Storage Module Commands 4-7

5. TELECOMMUNICATIONS

5.1 Telecommunications Commands 5-1
5.2 Remote Programming of the CR10 5-4

6. 9-PIN SERIAL INPUT/OUTPUT

6.1 Pin Description 6-1
6.2 Enabling and Addressing Peripherals 6-2
6.3 Ring Interrupts 6-3
6.4 Interrupts During Data Transfer 6-3
6.5 Modem/Terminal Peripherals 6-4
6.6 Synchronous Device Communication 6-4
6.7 Modem/Terminal and Computer Requirements 6-5

PROGRAM EXAMPLES

7. MEASUREMENT PROGRAMMING EXAMPLES

7.1 Single-Ended Voltage - LI200S Silicon Pyranometer 7-1
7.2 Differential Voltage Measurement 7-2
7.3 Thermocouple Temperatures Using the Optional CR10TCR to Measure the Reference Temperature 7-3
7.4 Thermocouple Temperatures Using an External Reference Junction 7-3
7.5 107 Temperature Probe 7-4
7.6 207 Temperature and RH Probe 7-4
7.7 Anemometer with Photochopper Output 7-5
7.8 Tipping Bucket Rain Gage with Long Leads 7-6
7.9 100 ohm PRT in 4 Wire Half Bridge 7-6
7.10 100 ohm PRT in 3 Wire Half Bridge 7-8
7.11 100 ohm PRT in 4 Wire Full Bridge 7-9
7.12 Pressure Transducer - 4 Wire Full Bridge 7-10
7.13 Lysimeter - 6 Wire Full Bridge 7-11
7.14 227 Gypsum Soil Moisture Block 7-13
7.15 Nonlinear Thermistor in Half Bridge (Model 101 Probe) 7-14
7.16 Water Level - Geokon's Vibrating Wire Pressure Sensor 7-15
7.17 Paroscientific "T" Series Pressure Transducer 7-19
7.18 SDM Peripherals 7-24
7.19 Paroscientific Pressure Transducer Processing 7-24

8. PROCESSING AND PROGRAM CONTROL EXAMPLES

8.1 Computation of Running Average 8-1
8.2 Rainfall Intensity 8-2
8.3 Using Control Ports and Loop to Run AM416 Multiplexer 8-3
8.4 Sub 1 Minute Output Interval Synched to Real Time 8-5
8.5 Interrupt Subroutine Used to Count Switch Closures (Rain Gage) 8-5
8.6 SDM-A04 Analog Output Multiplexer to Strip Chart 8-7
8.7 Converting 0-360 Wind Direction Output to 0-540 for Strip Chart 8-8
8.8 Use of 2 Final Storage Areas - Saving Data Prior to Event 8-9
8.9 Logarithmic Sampling Using Loops 8-10

INSTRUCTIONS

9. INPUT/OUTPUT INSTRUCTIONS 9-1

10. PROCESSING INSTRUCTIONS 10-1

11. OUTPUT PROCESSING INSTRUCTIONS 11-1

12. PROGRAM CONTROL INSTRUCTIONS 12-1

MEASUREMENTS

13. CR10 MEASUREMENTS

13.1 Fast and Slow Measurement Sequence 13-1
13.2 Single-Ended and Differential Voltage Measurements 13-2
13.3 The Effect of Sensor Lead Length on the Signal Settling Time 13-3
13.4 Thermocouple Measurements 13-12
13.5 Bridge Resistance Measurements 13-17
13.6 Resistance Measurements Requiring AC Excitation 13-21
13.7 Calibration Process 13-22

INSTALLATION

14. INSTALLATION AND MAINTENANCE

14.1 Protection from the Environment 14-1
14.2 Power Requirements 14-1
14.3 Campbell Scientific Power Supplies 14-2
14.4 Solar Panels 14-5
14.5 Direct Battery Connection to the CR10WP Wiring Panel 14-5
14.6 Vehicle Power Supply Connections 14-5
14.7 Grounding 14-6
14.8 Wiring Panel 14-7
14.9 Switched 12 Volt 14-7
14.10 Use of Digital I/O Ports for Switching Relays 14-7
14.11 Maintenance 14-9

APPENDICES

A. GLOSSARY A-1

B. CR10 PROM SIGNATURE AND OPTIONAL SOFTWARE

B.1 PROM Signature and Version B-1
B.2 Available PROMs/Library Options B-1
B.3 Description of Library Options Not in Standard Manual B-2

C. BINARY TELECOMMUNICATIONS

C.1 Telecommunications Command with Binary Responses C-1
C.2 Final Storage Format C-2
C.3 Generation of Signature C-4

D. CR10 37 PIN PORT DESCRIPTION D-1

E. ASCII TABLE E-1

G. CHANGING RAM OR PROM CHIPS

G.1 Disassembling the CR10 ..... G-1
G.2 Installing New RAM Chips in CR10s with 16K RAM G-1
G.3 Installing New PROM G-1
G.4 Installing 4K Program Memory PROM G-1

LIST OF TABLES LT-1

LIST OF FIGURES LF-1

INDEX I-1

SELECTED OPERATING DETAILS

1. **Storing Data** - Data are stored in Final Storage only by Output Processing Instructions and only when the Output Flag is set. (Sections OV4.1.1 and OV4.2.1)

2. **Storing Date and Time** - Date and time are stored with the data in Final Storage ONLY if the Real Time Instruction 77 is used. (Section 11)

3. **Data Transfer** - On-line data transfer from Final Storage to peripherals (printer, Storage Module, etc.) occurs only if enabled with Instruction 96 in the datalogger program. (Sections 4 and 12)

4. **Final Storage Resolution** - All Input Storage values are displayed (*6 mode) as high resolution with a maximum value of 99999. However, the default resolution for data stored in Final Storage is low resolution, maximum value of 6999. Results exceeding 6999 are stored as 6999 unless Instruction 78 is used to store the values in Final Storage as high resolution values. (Sections 2.2.1 and 11)

5. **Floating Point Format** - The computations performed in the CR10 use floating point arithmetic. CSI's 4 byte floating point numbers contain a 23 bit binary mantissa and a 6 bit binary exponent. The largest and smallest numbers that can be stored and processed are 9×10^{18} and 1×10^{-19}, respectively. (Section 2.2.2)

6. **Erasing Final Storage** - Data in Final Storage can be erased without altering the program by using the *A Mode to repartition memory. (Section 1.5.2)

7. **ALL memory can be erased** and the CR10 completely reset by entering 1986 for the number of bytes left in Program Memory. (Section 1.5.2)

8. **The set of instructions** available in the CR10 is determined by the PROM (Programmable Read Only Memory) that it is equipped with. Standard and optional software are identified in Appendix B. If you have ordered optional software that is not covered in the standard manual, the documentation is in Appendix H.

9. **Radiotelemetry Users** - As of February, 1990, CR10 PROMs no longer contain radio frequency interface software. That function is now contained in the RF95 Modem. To make measurements at a phone-to-RF base station using the RF100/RF200 Radio and RF95 Modem, current CR10 software is required. A CR10 with old software can be used with the new RF95 in the "RF95-ME" state, but the datalogger loses the "callback" capability as well as the SDC function.

10. **Changes with the release of OS10-0.1**:

    Wind Vector Instruction 69 has replaced Instruction 76. The options to do sub-interval averaging of the standard deviation of wind direction, $\sigma(\theta)$, and to calculate $\sigma(\theta)$ using the Yamartino algorithm have been added to the previous options (Section 9).

    Intermediate Processing Disable Flag 9 in now set low if a conditional test for setting it high fails (same as Output Flag 0, Section 3.7.2).

    ***D options for saving and loading programs with a cassette tape** are no longer in a standard PROM and must be ordered as a library option PROM (Appendix B).

CAUTIONARY NOTES

1. Damage will occur to the analog input circuitry if voltages in excess of ±16 V are applied for a sustained period. Voltages in excess of ±5V will cause errors and possible overranging on other analog input channels.

2. When using the CR10 with the PS12LA, remember that the sealed lead acid batteries are permanently damaged if discharged below 10.5 V. The cells are rated at a 7 Ahr capacity but experience a slow discharge even in storage. It is advisable to maintain a continuous charge on the PS12LA battery pack, whether in operation or storage (Section 14).

3. When connecting power to the CR10, first connect the positive lead from the power source to the 12 V terminal. Then connect the negative lead to G. Connecting these leads in the reverse order creates the possibility of a short (Section 14).

4. There are frequent references in this manual to Storage Modules. The Storage Modules referred to are the SM192 and SM716. The old SM16 and SM64 Storage Modules will NOT work with the CR10 without a specially modified cable. In addition, the SM16 and SM64 cannot perform many of the functions that the SM192 and SM716 are capable of performing.

5. Voltages in excess of 5.5 volts applied to a control port can cause the CR10 to malfunction.

6. Voltage pulses can be counted by CR10 Pulse Counters configured for High Frequency Pulses. However, when the pulse is actually a low frequency signal (below about 10 Hz) AND the positive voltage excursion exceeds 5.6 VDC, the 5 VDC supply will start to rise, upsetting all analog measurements.

   Pulses whose positive voltage portion exceed 5.6 VDC with a duration longer than 100 milliseconds need external conditioning. See the description of the Pulse count instruction in Section 9 for details on the external conditioning.

7. The CR10 module is sealed and contains desiccant to protect against excess humidity. The Wiring Panel and the connections between the Wiring Panel and the CR10 are still susceptible to humidity. To prevent corrosion at these points, additional desiccant must be placed inside the enclosure. To reduce vapor transfer into the enclosure, plug the cable entry conduit with Duct Seal, a putty-type sealant available at most electrical supply houses. DO NOT totally seal enclosures equipped with lead acid batteries. Hydrogen concentration may build up to explosive levels.

CR10 MEASUREMENT AND CONTROL MODULE OVERVIEW

Campbell Scientific Inc. provides four aids to understanding and operating the CR10:

1. *PCTOUR*
2. *This Overview*
3. *The CR10 Operator's Manual*
4. *The CR10 Prompt Sheet*

PCTOUR is a computer-guided tour of CR10 operation and the use of the PC208 Datalogger Support Software. Much of the material in this Overview is covered in PCTOUR. A copy of PCTOUR is included with every datalogger or PC208 order.

This Overview introduces the concepts required to take advantage of the CR10's capabilities. Hands-on programming examples start in Section OV5. Working with a CR10 will help the learning process, so don't just read the examples, do them. If you want to start this minute, go ahead and try the examples, then come back and read the rest of the Overview.

The sections of the Operator's Manual which should be read to complete a basic understanding of the CR10 operation are the Programming Sections 1-3, the portions of the data retrieval Sections 4 and 5 appropriate to the method(s) you are using (see OV6), and Section 14 which covers installation and maintenance.

Section 6 covers details of serial communications. Sections 7 and 8 contain programming examples. Sections 9-12 have detailed descriptions of each programming instruction, and Section 13 goes into detail on the CR10 measurement procedures.

The Prompt Sheet is an abbreviated description of the programming instructions. Once familiar with the CR10, it is possible to program it using only the Prompt Sheet as a reference, consulting the manual if further detail is needed.

Read the Selected Operating Details and Cautionary Notes at the front of the Manual before using the CR10.

OV1. PHYSICAL DESCRIPTION

The CR10 is a fully programmable datalogger/controller in a small, rugged, sealed module. Programming is very similar to Campbell Scientific's 21X and CR7 dataloggers. Some fundamental physical differences are listed below.

- The CR10 does not have an integral keyboard/display. The user accesses the CR10 with the portable CR10KD Keyboard Display or with a computer or terminal (Section OV2).
- The CR10 does not have an integral terminal strip. A removable wiring panel (Figure OV1.1-1) performs this function and attaches to the two D-type connectors located at the end of the module.
- The power supply is external to the CR10. This gives the user a wide range of options (Section 14) for powering the CR10.

OV1.1 WIRING PANEL

The CR10 Wiring Panel and CR10 datalogger make electrical contact through the two D-type connectors at the (left) end of the CR10.

The Wiring Panel contains a 9-pin Serial I/O port used when communicating with the datalogger and provides terminals for connecting sensor, control, and power leads to the CR10. It also provides transient protection and reverse polarity protection. Figure OV1.1-2 shows the panel and the instructions used to access the various terminals.

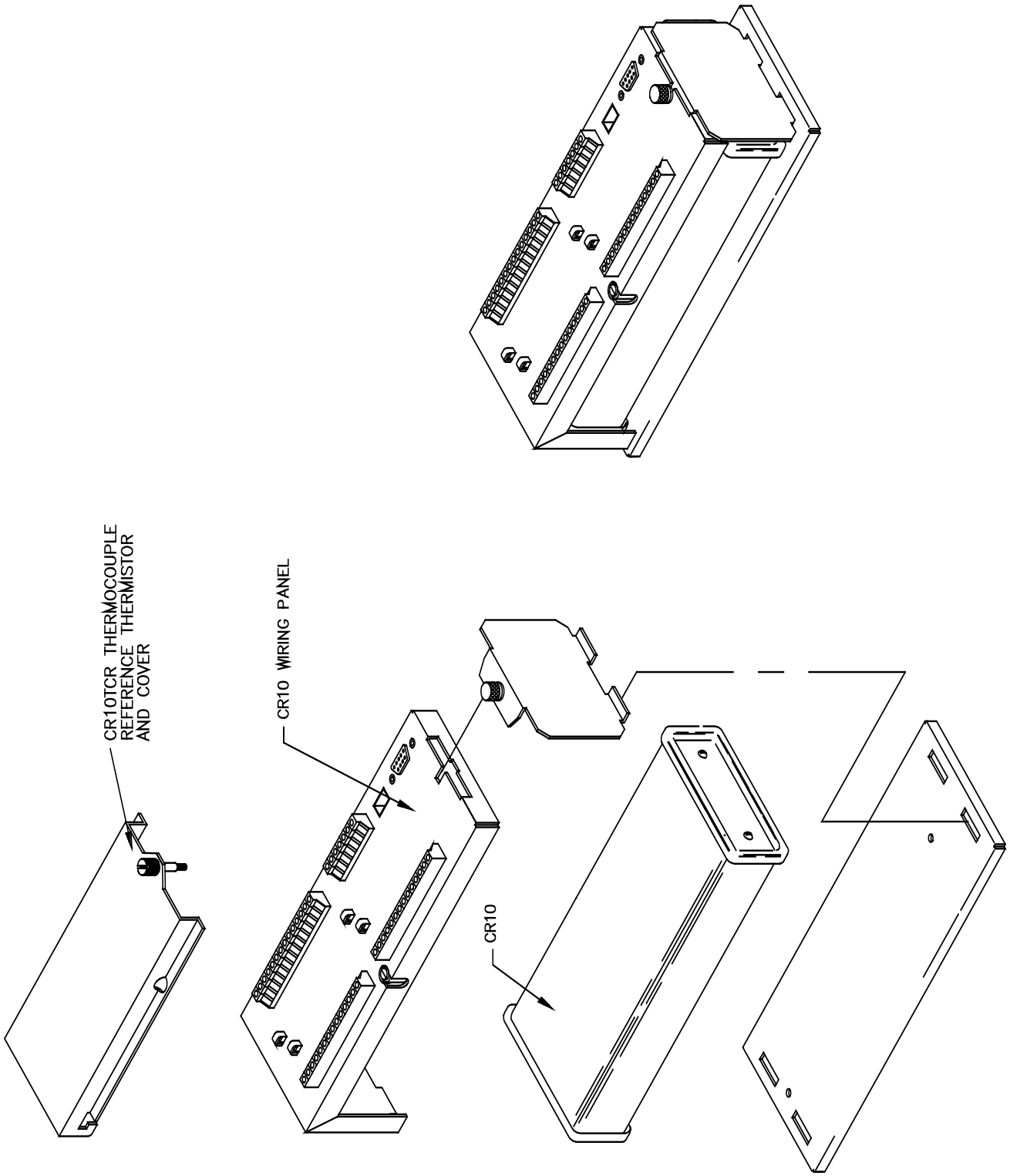

CR10TCR THERMOCOUPLE
REFERENCE THERMISTOR
AND COVER
CR10 WIRING PANEL
CR10

FIGURE OV1.1-1. CR10 and Wiring Panel

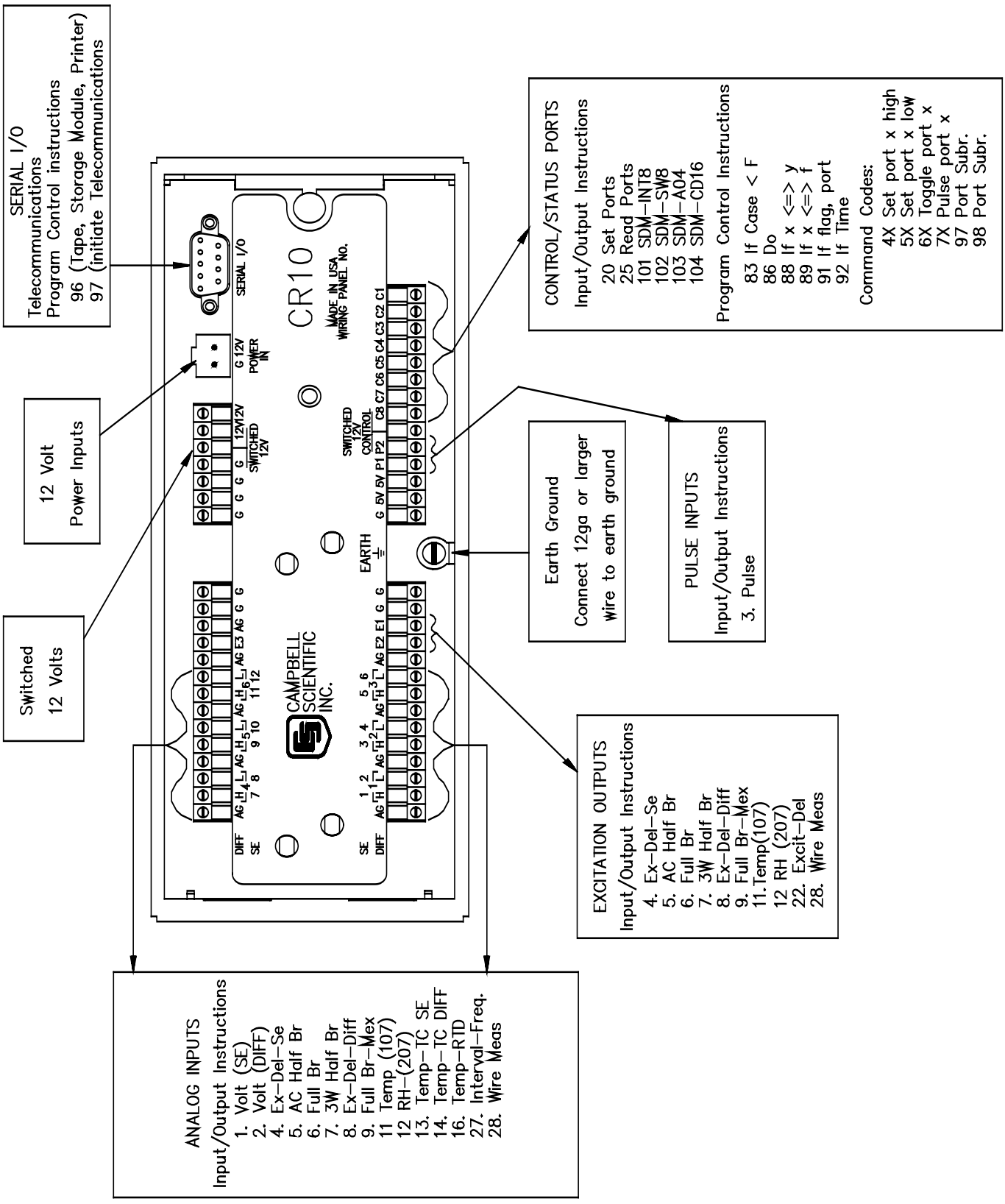


FIGURE OV1.1-2. CR10 Wiring Panel/Instruction Access

OV1.1.1 ANALOG INPUTS

The terminals labeled 1H to 6L are analog inputs. These numbers refer to the high and low inputs to the differential channels 1 through 6. In a differential measurement, the voltage on the H input is measured with respect to the voltage on the L input. When making single-ended measurements, either the H or L input may be used as an independent channel to measure voltage with respect to the CR10 analog ground (AG). The single-ended channels are numbered sequentially starting with 1H; e.g., the H and L sides of differential channel 1 are single-ended channels 1 and 2; the H and L sides of differential channel 2 are single-ended channels 3 and 4, etc. (The single-ended channel numbers do NOT appear on older wiring panels).

OV1.1.2 SWITCHED EXCITATION OUTPUTS

The terminals labeled E1, E2, and E3 are precision, switched excitation outputs used to supply programmable excitation voltages for resistive bridge measurements. DC or AC excitation at voltages between -2500 mV and +2500 mV are user programmable (Section 9).

OV1.1.3 PULSE INPUTS

The terminals labeled P1 and P2 are the pulse counter inputs for the CR10. They are programmable for switch closure, high frequency pulse or low level AC (Section 9, Instruction 3).

OV1.1.4 DIGITAL I/O PORTS

Terminals C1 through C8 are digital Input/Output ports. On power-up they are configured as input ports, commonly used for reading the status of an external signal. High and low conditions are: 3V < high < 5.5V; -0.5V < low < 0.8V.

Configured as outputs the ports allow on/off control of external devices. A port can be set high (5V ± 0.1V), set low (<0.1V), toggled or pulsed (Sections 3, 8.3, and 12).

OV1.1.5 ANALOG GROUND (AG)

The AG terminals are analog grounds, used as the reference for single-ended measurements and excitation return.

OV1.1.6 12V AND POWER GROUND (G) TERMINALS

The 12V and power ground (G) terminals are used to supply 12V DC power to the datalogger. The extra 12V and G terminals can be used to connect other devices requiring 12V power.

The G terminals are also used to tie cable shields to ground, and to provide a ground reference for pulse counters and binary inputs. For protection against transient voltage spikes, power ground should be connected to a good earth ground (Section 14.3.1).

OV1.1.7 5V OUTPUTS

The two 5V (±0.2%) outputs are commonly used to power peripherals such as the QD1 Incremental Encoder Interface, AVW1 or AVW4 Vibrating Wire Interface.

The 5V outputs are common with pin 1 on the 9 pin serial connector; 200 mA is the maximum combined output.

OV1.1.8 SERIAL I/O

The 9 pin serial I/O port contains lines for serial communication between the CR10 and external devices such as computers, printers, Storage Modules, etc. **This port does NOT have the same configuration as the 9 pin serial ports currently used on many personal computers**. It has a 5VDC power line which is used to power peripherals such as the SM192 or SM716 Storage Module or the DC112 Phone Modem. The same 5VDC supply is used for the 5V outputs on the lower terminal strip. Section 6 contains technical details on serial communication.

OV1.1.9 SWITCHED 12 VOLT

Wiring panels introduced in March 1994 include a switched 12 volt output. This can be used to power sensors or devices requiring an unregulated 12 volts. The output is limited to 600 mA current.

A control port is used to operate the switch. Connect a wire from the control port to the switched 12 volt control port. When the port is set high, the 12 volts is turned on; when the port is low, the switched 12 volts is off.

OV1.2 CONNECTING POWER TO THE CR10

The CR10 can be powered by any 12VDC source. First connect the positive lead from the power supply to one of the 12V terminals and then connect the negative lead to one of the power ground (G) terminals. The Wiring Panel power connection is reverse polarity protected. See Section 14 for details on power supply connections.

> **CAUTION**: The metal surfaces of the CR10 Wiring Panel, and CR10KD Keyboard Display are at the same potential as power ground. To avoid shorting 12 volts to ground, connect the 12 volt lead first, then connect the ground lead.

OV2. MEMORY AND PROGRAMMING CONCEPTS

The CR10 must be programmed before it will make any measurements. A program consists of a group of instructions entered into a **program table**. The program table is given an **execution interval** which determines how frequently that table is executed. When the table is executed, the instructions are executed in sequence from beginning to end. After executing the table, the CR10 waits the remainder of the execution interval and then executes the table again starting at the beginning.

The interval at which the table is executed generally determines the interval at which the sensors are measured. The interval at which data are stored is separate from how often the table is executed, and may range from samples every execution interval to processed summaries output hourly, daily, or on longer or irregular intervals.

Figure OV2.1-1 represents the measurement, processing, and data storage sequence, and the types of instructions used to accomplish these tasks.

OV2.1 INTERNAL MEMORY

The CR10 has 64K bytes of Random Access Memory (RAM), divided into five areas. The use of the Input, Intermediate, and Final Storage in the measurement and data processing sequence is shown in Figure OV2.1-1. While the total size of these three areas remains constant, memory may be reallocated between the areas to accommodate different measurement and processing needs (*A Mode, Section 1.5). The size of the 2 additional memory areas, system and program, are fixed. The five areas of RAM are:

1. **Input Storage** - Input Storage holds the results of measurements or calculations. The *6 Mode is used to view Input Storage locations for checking current sensor readings or calculated values. Input Storage defaults to 28 locations. Additional locations can be assigned using the *A Mode (Section 1.5).

2. **Intermediate Storage** - Certain Processing Instructions and most of the Output Processing Instructions maintain intermediate results in Intermediate Storage. Intermediate storage is automatically accessed by the instructions and cannot be accessed by the user. The default allocation is 64 locations. The number of locations can be changed using the *A Mode.

3. **Final Storage** - Final processed values are stored here for transfer to printer, solid state Storage Module or for retrieval via telecommunication links. Values are stored in Final Storage only by the Output Processing Instructions and only when the Output Flag is set in the users program. Approximately 29,900 locations are allocated to Final Storage on power up. This number is reduced if Input or Intermediate Storage is increased.

4. **System Memory** - used for overhead tasks such as compiling programs, transferring data etc. The user cannot access this memory.

5. **Program Memory** - available for user programs entered in program tables 1 and 2, and Subroutine Table 3.

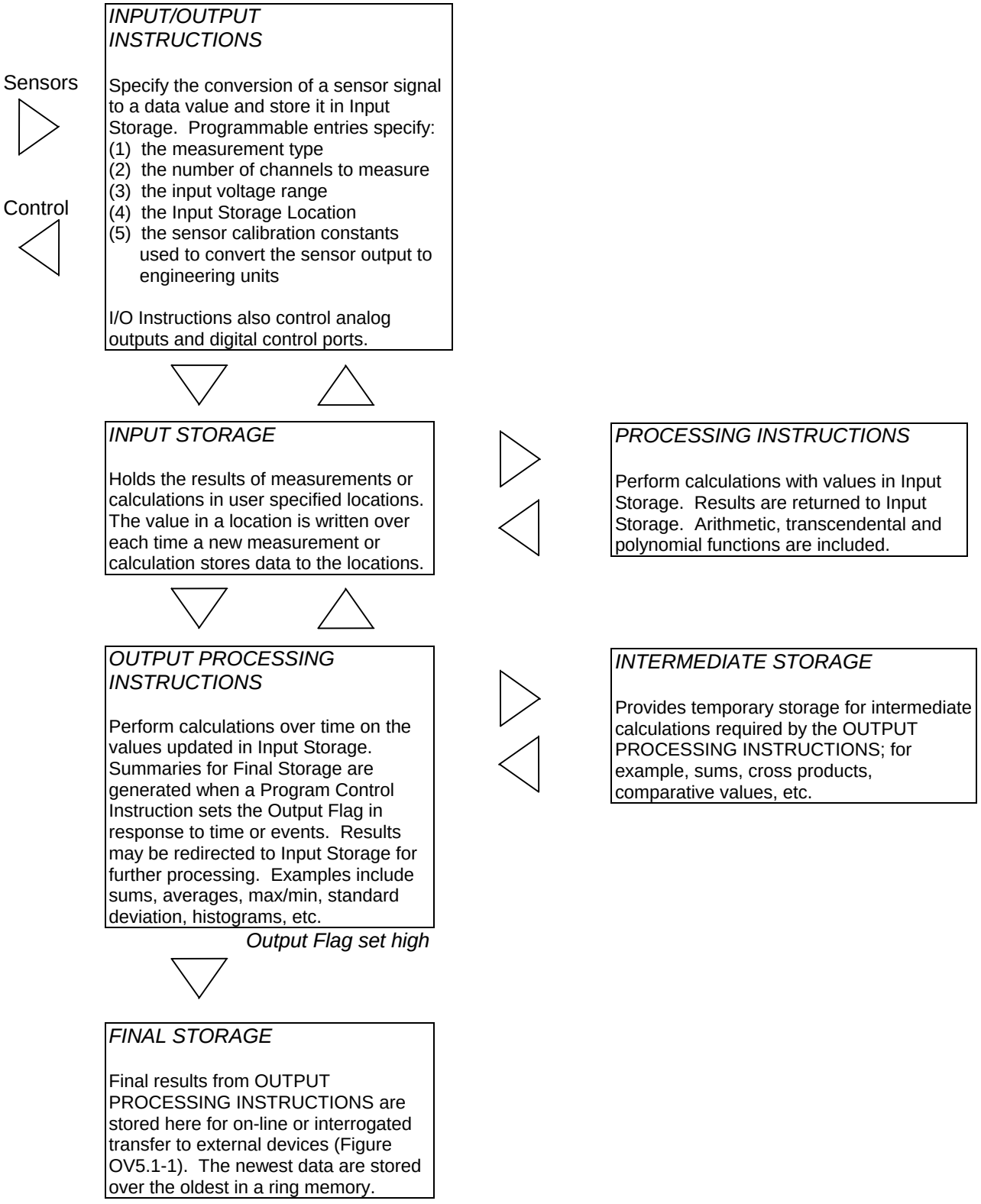

Sensors
Control
INPUT/OUTPUT INSTRUCTIONS
Specify the conversion of a sensor signal to a data value and store it in Input Storage. Programmable entries specify:
(1) the measurement type
(2) the number of channels to measure
(3) the input voltage range
(4) the Input Storage Location
(5) the sensor calibration constants used to convert the sensor output to engineering units
I/O Instructions also control analog outputs and digital control ports.
INPUT STORAGE
Holds the results of measurements or calculations in user specified locations. The value in a location is written over each time a new measurement or calculation stores data to the locations.
PROCESSING INSTRUCTIONS
Perform calculations with values in Input Storage. Results are returned to Input Storage. Arithmetic, transcendental and polynomial functions are included.
OUTPUT PROCESSING INSTRUCTIONS
Perform calculations over time on the values updated in Input Storage. Summaries for Final Storage are generated when a Program Control Instruction sets the Output Flag in response to time or events. Results may be redirected to Input Storage for further processing. Examples include sums, averages, max/min, standard deviation, histograms, etc.
INTERMEDIATE STORAGE
Provides temporary storage for intermediate calculations required by the OUTPUT PROCESSING INSTRUCTIONS; for example, sums, cross products, comparative values, etc.
Output Flag set high
FINAL STORAGE
Final results from OUTPUT PROCESSING INSTRUCTIONS are stored here for on-line or interrogated transfer to external devices (Figure OV5.1-1). The newest data are stored over the oldest in a ring memory.

FIGURE OV2.1-1. Instruction Types and Storage Areas

OV2.2 CR10 INSTRUCTION TYPES

Figure OV2.1-1 illustrates the use of three different instruction types which act on data. The fourth type, Program Control, is used to control output times and vary program execution. Instructions are identified by numbers.

1. **INPUT/OUTPUT INSTRUCTIONS** (1-28, 101-104, Section 9) control the terminal strip inputs and outputs (the sensor is the source, Figure OV1.1-2), storing the results in Input Storage (destination). Multiplier and offset parameters allow conversion of linear signals into engineering units. The Digital I/O Ports are also addressed with I/O Instructions.

2. **PROCESSING INSTRUCTIONS** (30-66, Section 10) perform numerical operations on values located in Input Storage (source) and store the results back in Input Storage (destination). These instructions can be used to develop high level algorithms to process measurements prior to Output Processing.

3. **OUTPUT PROCESSING INSTRUCTIONS** (69-82, Section 11) are the only instructions which store data in Final Storage (destination). Input Storage (source) values are processed over time to obtain averages, maxima, minima, etc. There are two types of processing done by Output Instructions: **Intermediate** and **Final**.

   Intermediate processing normally takes place each time the instruction is executed. For example, when the Average Instruction is executed, it adds the values from the input locations being averaged to running totals in Intermediate Storage. It also keeps track of the number of samples.

   Final processing occurs only when the Output Flag is high. The Output Processing Instructions check the Output Flag. If the flag is high, final values are calculated and output. With the Average, the totals are divided by the number of samples and the resulting averages sent to Final Storage. Intermediate locations are zeroed and the process starts over. *The Output Flag, Flag 0, is set high by a Program Control Instruction which must precede the Output Processing Instructions in the user entered program.*

4. **PROGRAM CONTROL INSTRUCTIONS** (83-98, Section 12) are used for logic decisions and conditional statements. They can set flags, compare values or times, execute loops, call subroutines, conditionally execute portions of the program, etc.

OV2.3 PROGRAM TABLES, EXECUTION INTERVAL AND OUTPUT INTERVALS

Programs are entered in Tables 1 and 2. Subroutines, called from Tables 1 and 2, are entered in Subroutine Table 3. The size of each table is flexible, limited only by the total amount of program memory. If Table 1 is the only table programmed, the entire program memory is available for Table 1.

Table 1 and Table 2 have independent execution intervals, entered in units of seconds with an allowable range of 1/64 to 8191 seconds. Subroutine Table 3 has no execution interval; subroutines are only executed when called from Table 1 or 2.

OV2.3.1 THE EXECUTION INTERVAL

The execution interval specifies how often the program in the table is executed, which is usually determined by how often the sensors are to be measured. *Unless two different measurement rates are needed, use only one table*. A program table is executed sequentially starting with the first instruction in the table and proceeding to the end of the table.

Table 1. Execute every x sec. 0.0156 ≤ x ≤ 8191
Instructions are executed sequentially in the order they are entered in the table. One complete pass through the table is made each execution interval unless program control instructions are used to loop or branch execution.
Normal Order: MEASURE PROCESS CHECK OUTPUT COND. OUTPUT PROCESSING

Table 2. Execute every y sec. 0.0156 ≤ y ≤ 8191
Table 2 is used if there is a need to measure and process data on a separate interval from that in Table 1.

Table 3. Subroutines
A subroutine is executed only when called from Table 1 or 2.
Subroutine Label Instructions End
Subroutine Label Instructions End
Subroutine Label Instructions End

FIGURE OV2.3-1. Program and Subroutine Tables

Each instruction in the table requires a finite time to execute. If the execution interval is less than the time required to process the table, an execution interval overrun occurs; the CR10 finishes processing the table and waits for the next execution interval before initiating the table. When an overrun occurs, decimal points are shown on either side of the G on the display in the LOG mode (*0). Overruns and table priority are discussed in Section 1.1.

OV2.3.2. THE OUTPUT INTERVAL

The interval at which output occurs is independent from the execution interval, other than the fact that it must occur when the table is executed (e.g., a table cannot have a 10 minute execution interval and output every 15 minutes).

A single program table can have many different output intervals and conditions, each with a unique data set (Output Array). Program Control Instructions are used to set the Output Flag. The Output Processing Instructions which follow the instruction setting the Output Flag determine the data output and its sequence. Each additional Output Array is created by another Program Control Instruction checking a output condition, followed by Output Processing Instructions defining the data set to output.

OV3. COMMUNICATING WITH CR10

An external device must be connected to the CR10's Serial I/O port to communicate with the CR10. This may be either Campbell Scientific's portable CR10KD Keyboard Display or a computer/terminal.

The CR10KD is powered by the CR10 and connects directly to the serial port via the SC12 cable (supplied with the CR10KD). No interfacing software is required.

To communicate with any device other than the CR10KD, the CR10 enters its Telecommunications Mode and responds only to valid telecommunications commands. Within the Telecommunications Mode, there are 2 "states"; the Telecommunications Command state and the Remote Keyboard state. Communication is established in the Telecommunications command state. One of the commands is to enter the Remote Keyboard state.

The Remote Keyboard state allows the keyboard of the computer/terminal to act like the CR10KD keyboard. Various datalogger modes may be entered, including the mode in which programs may be keyed in to the CR10 from the computer/terminal.

Campbell Scientific's PC208 Datalogger Support Software facilitates the use of IBM PC/XT/AT/PS-2's and compatibles for communicating with the CR10. This package

contains a program editor (EDLOG), a terminal emulator (GraphTerm), telecommunications (TELCOM), a data reduction program (SPLIT), and programs to retrieve data from both generations of Campbell Scientific's Storage Modules (SMREAD and SMCOM).

To participate in the programming examples (Section OV5) you must communicate with the CR10. Read Section OV3.1 if the CR10KD is being used, Section OV3.2 if the PC208 software is being used, or Section 3.3 and Section 5 if some other computer or terminal is being used.

OV3.1 CR10 KEYBOARD/DISPLAY

The SC12 cable (supplied with the CR10KD) is used to connect the Keyboard/Display to the 9 pin Serial I/O port on the CR10.

If the Keyboard/Display is connected to the CR10 prior to being powered up, the "HELLO" message is displayed while the CR10 checks memory. The size of the usable system memory is then displayed (96 for 96K bytes of memory). When the CR10KD is plugged in after the CR10 has powered up, the display is meaningless until "*" is pressed to enter a mode.

OV3.2 USING THE PC208 TERMINAL EMULATOR (GRAPHTERM)

For IBM compatible computers, the PC208 software contains a terminal emulator program called GraphTerm. When using GraphTerm, the baud rate, port, and modem types are specified and stored in a file for future use.

The simplest and most common interface is the SC32A Optically Isolated RS232 Interface. The SC32A converts and optically isolates the voltages passing between the CR10 and the external terminal device.

The SC12 Two Peripheral cable which comes with the SC32A is used to connect the serial I/O port of the CR10 to the 9 pin port of the SC32A labeled "Datalogger". Connect the "Terminal/Printer" port of the SC32A to the serial port of the computer with a straight 25 pin cable or, if the computer has a 9 pin serial port, a standard 9 to 25 pin adapter cable.

To establish the communication link between the computer and the CR10, the user may either select the T option and send carriage returns as described above or select the "C" option to "Call" the station (see PC208 Operator's Manual). Once the link is active, issue the "7H" command to enter the Remote Keyboard State.

OV3.3 ASCII TERMINAL OR COMPUTER WITH TERMINAL EMULATOR

Devices which can be used to communicate with the CR10 include standard ASCII terminals and computers programmed to function as a terminal emulator.

OV3.3.1 COMPUTER/TERMINAL REQUIREMENTS

The basic requirements are:

1. There must be an asynchronous serial port to transmit and receive characters.
2. Communication protocol must be matched for the two devices.
3. The proper cable/interface must be used between the serial ports.
4. A computer must be programmed to function as a terminal.

While the connection between the computer/terminal and the CR10 may be via modem (phone, RF, or short haul), the most frequently used device for a short connection is the SC32A Optically Isolated RS232 Interface.

Most computer/terminal devices require RS232 input logic levels of -5V for logic low and +5V for logic high. Logic levels from the CR10's serial I/O port are 0V for logic low and +5V for logic high.

The SC32A converts and optically isolates the voltages passing between the CR10 and the external terminal device. The SC32A is configured as Data Communications Equipment (DCE) for direct connection to Data Terminal Equipment (DTE) which includes most computers and terminals.

The SC12 Two Peripheral cable which comes with the SC32A is used to connect the serial I/O port of the CR10 to the 9 pin port of the SC32A labeled "Datalogger". Connect the "Terminal/Printer" port of the SC32A to the serial port of the terminal with a user supplied

straight cable with the proper connectors (Campbell Scientific SC25PS or equivalent for a 25 pin serial port configured DTE).

OV3.3.2 ESTABLISHING COMMUNICATION WITH THE CR10

Communication software is available for most computers having a serial port. Campbell Scientific's PC208 Datalogger Support Software is available for IBM PC/XT/AT/PS-2's and compatibles. The software must be capable of the following communication protocol:

1. Configuring an asynchronous serial port for 8 Data Bits, 1 Stop Bit, no Parity, and Full Duplex at baud rates of 300, 1200, or 9600 baud.
2. Transmitting characters typed on the keyboard out through the serial port.
3. Displaying characters/data received through the computer's serial port.

Once the computer is functioning as a terminal, initiate communications by sending the CR10 several carriage returns for the CR10 to match the baud rate and respond with "*". Enter the 7H command to enter the Remote Keyboard State. At this point, the CR10 can be controlled using the Keyboard Commands described in Section OV4. For additional information on communications, see Section 6.7.

OV4. PROGRAMMING THE CR10

A program is created by entering it directly into the datalogger or on a computer using the PC208 Datalogger Support Software program EDLOG. This manual describes direct interaction with the CR10. Work through the direct programming examples in this overview before using EDLOG and you will have the basics of CR10 operation as well as an appreciation for the help provided by the software. Section OV4.5 describes options for loading the program into the CR10.

OV4.1 FUNCTIONAL MODES

CR10/User interaction is broken into different functional MODES (e.g., programming the measurements and output, setting time, manually initiating a block data transfer to Storage Module, etc.). The modes are referred to as Star (*) Modes since they are accessed by first keying *, then the mode number or letter. Table OV4.1-1 lists the CR10 Modes.

TABLE OV4.1-1. * Mode Summary

Key	Mode
*0	LOG data and indicate active Tables
*1	Program Table 1
*2	Program Table 2
*3	Program Table 3, subroutines only
*5	Display/set real time clock
*6	Display/alter Input Storage data, toggle flags or control ports.
*7	Display Final Storage data
*8	Final Storage data transfer to peripheral
*9	Storage Module commands
*A	Memory allocation/reset
*B	Signature/status
*C	Security
*D	Save/load Program

OV4.2 KEY DEFINITION

Keys and key sequences have specific functions when using the CR10KD keyboard or a computer/terminal in the remote keyboard state (Section 5). Table OV4-2 lists these functions. In some cases, the exact action of a key depends on the mode the CR10 is in and is described with the mode in the manual.

TABLE OV4.2-1. Key Description/Editing Functions

Key	Action
0-9	Key numeric entries into display
*	Enter Mode (followed by Mode Number)
A	Enter/Advance
B	Back up
C	Change the sign of a number or index an input location to loop counter
D	Enter the decimal point
#	Clear the rightmost digit keyed into the display
#A	Advance to next instruction in program table (*1, *2, *3) or to next Output Array in Final Storage (*7)
#B	Back up to previous instruction in program table or to previous Output Array in Final Storage
#D	Delete entire instruction
#0	(then A or CR) Back up to the start of the current array.

When using a computer/terminal to communicate with the CR10 (Telecommunications) there are some keys available in addition to those found on the CR10KD. Table OV4.2-2 lists these keys.

TABLE OV4.2-2. Additional Keys Allowed in Telecommunications

Key	Action
-	Change Sign, Index (same as C)
CR	Enter/advance (same as A)
:	Colon (used in setting time)
S or ^S	Stops transmission of data (10 second time-out; any character restarts)
C or ^C	Aborts transmission of Data

OV4.3 PROGRAMMING SEQUENCE

In routine applications, the CR10 measures sensor output signals, processes the measurements over some time interval and stores the processed results. A generalized programming sequence is:

1. Enter the execution interval. In most cases, the execution interval is determined by the desired sensor scan rate.

2. Enter the Input/Output instructions required to measure the sensors.

3. If processing in addition to that provided by the Output Processing Instructions (step 5) is required, enter the appropriate Processing Instructions.

4. Enter the Program Control Instruction to test the output condition and set the Output Flag when the condition is met. For example, use

   Instruction 92 to output based on time.

   Instruction 86 to output every execution interval.

   Instruction 88 or 89 to output based on a comparison of values in input locations.

   This instruction must precede the Output Processing Instructions which store data in Final Storage. Instructions are described in Sections 9 through 12.

5. Enter the Output Processing Instructions to store processed data in Final Storage. The order in which data are stored is

determined by the order of the Output Processing Instructions in the table.

6. Repeat steps 4 through 6 for additional outputs on different intervals or conditions.

> **NOTE**: The program must be executed for output to occur. Therefore, the interval at which the Output Flag is set must be evenly divisible by the execution interval. For example, with a 2 minute execution interval and a 5 minute output interval, the program will only be executed on the even multiples of the 5 minute intervals, not on the odd. Data will be output every 10 minutes instead of every 5 minutes.

Execution intervals and output intervals set with Instruction 92 are synchronized with real time starting at midnight.

OV4.4 INSTRUCTION FORMAT

Instructions are identified by an instruction number. Each instruction has a number of parameters that give the CR10 the information it needs to execute the instruction.

The CR10 Prompt Sheet has the instruction numbers in red, with the parameters briefly listed in columns following the description. Some parameters are footnoted with further description under the "Instruction Option Codes" heading.

For example, Instruction 73 stores the maximum value that occurred in an Input Storage location over the output interval. The instruction has three parameters (1) REPetitionS, the number of sequential Input Storage locations on which to find maxima, (2) TIME, an option of storing the time of occurrence with the maximum value, and (3) LOC the first Input Storage location operated on by the Maximum Instruction. The codes for the TIME parameter are listed in the "Instruction Option Codes".

The repetitions parameter specifies how many times an instruction's function is to be repeated. For example, four 107 thermistor probes may be measured with a single Instruction 11, Temp-107, with four repetitions. Parameter 2 specifies the input channel of the first thermistor (the probes must be connected to sequential channels). Parameter 4 specifies the Input Storage location in which to store measurements from the first thermistor. If location 5 were used and the first probe was on channel 1, the temperature of the thermistor on channel 1 would be stored in input location 5, the temperature from channel 2 in input location 6, etc.

Detailed descriptions of the instructions are given in Sections 9-12. Entering an instruction into a program table is described in OV5.

OV4.5 ENTERING A PROGRAM

Programs are entered into the CR10 in one of three ways:

1. Keyed in using the CR10 keyboard.
2. Loaded from a pre-recorded listing using the *D Mode. There are 3 types of storage/input:
   a. Stored on disk/sent from computer (PC208 software GraphTerm and EDLOG).
   b. Stored/loaded from SM192/716 Storage Module.
3. Loaded from internal PROM (special software) or Storage Module upon power-up.

A program is created by keying it directly into the datalogger as described in Section OV5, or on a PC using the PC208 Datalogger Support Software.

EDLOG and GraphTerm are PC208 Software programs used to develop and send programs to Campbell Scientific dataloggers. EDLOG is an editor for writing and documenting programs for Campbell Scientific dataloggers. Program files developed with EDLOG can be downloaded directly to the CR10 using GraphTerm. GraphTerm supports communication via direct wire, telephone, or Radio Frequency (RF).

Programs on disk can be copied to a Storage Module with SMCOM. Using the *D Mode to save or load a program from a Storage Module is described in Section 1.8.

It is possible (with special software) to create a PROM (Programmable Read Only Memory) that contains a datalogger program. With this PROM installed in the datalogger, the program will automatically be loaded and run when the

datalogger is powered-up, requiring only that the clock be set.

The program on power up function can be achieved by using a SM192/716 Storage Module. Up to 8 programs can be stored in the Storage Module, the programs may be assigned any of the numbers 1-8. If the Storage Module is connected when the CR10 is powered-up the CR10 will automatically load program number 8, provided that a program 8 is loaded in the Storage Module (Section 1.8).

OV5. PROGRAMMING EXAMPLES

We will start with a simple programming example. There is a brief explanation of each step to help you follow the logic. When the example uses an instruction, find it on the Prompt Sheet and follow through the description of the parameters. Using the Prompt Sheet while going through these examples will help you become familiar with its format. Sections 9-12 have more detailed descriptions of the instructions.

Connect the CR10 to either a CR10KD Keyboard/Display or a terminal (Section OV2). With the Wiring Panel connected to the CR10, hook up the power leads as described in Section OV1.2. If using a terminal, use the 7H command to get into the Remote Keyboard State (Sections 5.2). The programming steps in the following examples use the keystrokes possible on the keyboard/display. With a terminal, some responses will be slightly different.

If the CR10KD is connected to the CR10 when it is powered up, the display will show:

Display	Explanation
HELLO	On power-up, the CR10 displays "HELLO" while it checks the memory (this display occurs only with the CR10KD).

after a few seconds delay

:96	The size of the machine's total memory (RAM plus 32 K of ROM), in this case 96K

OV5.1 SAMPLE PROGRAM 1

In this example the CR10 is programmed to read its own internal temperature (using a built in thermistor) every 5 seconds and to send the results to Final Storage.

Key	Display Will Show: (ID:Data)	Explanation
*	00:00	Enter mode.
1	01:00	Enter Program Table 1.
A	01:0.0000	Advance to execution interval (In seconds)
5	01:5	Key in an execution interval of 5 seconds.
A	01:P00	Enter the 5 second execution interval and advance to the first program instruction location.
17	01:P17	Key in Instruction 17 which directs the CR10 to measure the internal temperature in degrees C. This is an Input/Output Instruction.
A	01:0000	Enter Instruction 17 and advance to the first parameter.
1	01:1	The input location to store the measurement, location 1.
A	02:P00	Enter the location # and advance to the second program instruction.

The CR10 is now programmed to read the internal temperature every 5 seconds and place the reading in Input Storage Location 1. The program can be compiled and the temperature displayed.

Key	Display Will Show: (ID:Data)	Explanation
*0	LOG 1	Exit Table 1, enter *0 Mode, compile table and begin logging.
*6	06:0000	Enter *6 Mode (to view Input Storage).
A	01:21.234	Advance to first storage location. Panel temp. is 21.234ºC (display shows actual temp.).
	01:21.423	The CR10 has read the sensor and stored the result again. The internal temp is now 21.423 ºC. The value is updated every 5 seconds when the table is executed. At this point the CR10 is measuring the temperature every 5 seconds and sending the value to Input Storage. No data are being saved. The next step is to have the CR10 send each reading to Final Storage. (Remember, the Output Flag must be set first.)
*1	01:00	Exit *6 Mode. Enter program table 1.
2A	02:P00	Advance to 2nd instruction location (this is where we left off).
86	02:P86	This is the DO instruction (a Program Control Instruction).
A	01:00	Enter 86 and advance to the first parameter (which will specify the command to execute).
10	01:10	This command sets the Output Flag. (Flag 0)
A	03:P00	Enter 10 and advance to third program instruction.
70	03:P70	The SAMPLE instruction. It directs the CR10 to take a reading from an Input Storage location and send it to Final Storage (an Output Processing Instruction).
A	01:0000	Enter 70 and advance to the first parameter (repetitions).
1	01:1	There is only one input location to sample; repetitions = 1.

Wait a few seconds:

A	02:0000	Enter 1 and advance to second parameter (Input Storage location to sample).
1	02:1	Input Storage Location 1, where the temperature is stored.
A	04:P00	Enter 1 and advance to fourth program instruction.
*	00:00	Exit Table 1.
0	LOG 1	Enter *0 Mode, compile program, log data.

*The CR10 is now programmed to measure the internal temperature every 5 seconds and send each reading to Final Storage. Values in Final Storage can be viewed using the *7 Mode.*

Display Will Show:

Key	(ID:Data)	Explanation
*7	07: 13.000	Enter *7 Mode. The Data Storage Pointer (DSP) is at Location 13 (in this example).
A	01: 0102	Advance to the first value, the Output Array ID. 102 indicates the Output Flag was set by the second instruction in Program Table 1.
A	02: 21.23	Advance to the first stored temperature.
A	01: 0102	Advance to the next output array. Same Output Array ID.
A	02: 21.42	Advance to 2nd stored temp, 21.42 deg. C.

There are no date and time tags on the data. They must be put there with Output Instruction 77. Instruction 77 is used in the next example.

If a terminal is used to communicate with the CR10, Telecommunications Commands (Section 5) can be used to view entire Output Arrays (in this case the ID and temperature) at the same time.

OV5.2 SAMPLE PROGRAM 2

This second example is more representative of a real-life data collection situation. Once again the internal temperature is measured, but it is used as a reference temperature for the differential voltage measurement of a type T (copper-constantan) thermocouple; the CR10 should have arrived with a short type T thermocouple connected to differential channel 5.

When using a type T thermocouple, the copper lead (blue) is connected to the high input of the differential channel, and the constantan lead (red) is connected to the low input.

A thermocouple produces a voltage that is proportional to the difference in temperature between the measurement and the reference junctions.

To make a thermocouple (TC) temperature measurement, the temperature of the reference junction (in this example, the approximate panel temperature) must be measured. The CR10 takes the reference temperature, converts it to the equivalent TC voltage relative to 0^{o}C, adds the measured TC voltage, and converts the sum to temperature through a polynomial fit to the TC output curve (Section 13.4).

The internal temperature of the CR10 is not a suitable reference temperature for precision thermocouple measurements. It is used here for the purpose of training only. To make thermocouple measurements with the CR10, purchase the Campbell Scientific Thermocouple Reference, Model CR10TCR (Section 13.4) and make the reference temperature measurement with Instruction 11.

Instruction 14 directs the CR10 to make a differential TC temperature measurement. The first parameter in Instruction 14 is the number of times to repeat the measurement. Enter 1, because in this example there is only one thermocouple. If there were more than 1 TC, they could be wired to sequential channels, and the number of thermocouples entered for repetitions. The CR10 would automatically advance through the channels sequentially and measure all of the thermocouples.

Parameter 2 is the voltage range to use when making the measurement. The output of a type T thermocouple is approximately 40 microvolts per degree C difference in temperature between the two junctions. The ±2.5 mV scale will provide a range of +2500/40 = +62.5 ^{o}C (i.e., this scale will not overrange as long as the measuring junction is within 62.5 ^{o}C of the panel temperature). The resolution of the ±2.5 mV range is 0.33 µV or 0.008 ^{o}C.

Parameter 3 is the analog input channel on which to make the first, and in this case only, measurement.

Parameter 4 is the code for the type of thermocouple used. This information is located on the Prompt Sheet or in the description of Instruction 14 in Section 9. The code for a type T (copper-constantan) thermocouple is 1.

Parameter 5 is the Input Storage location in which the reference temperature is stored. Parameter 6 is the Input Storage location in which to store the measurement (or the first measurement; e.g., if there are 5 repetitions and the first measurement is stored in location 3, the final measurement will be stored in location 7). Parameters 7 and 8 are the multiplier and offset. A multiplier of 1 and an offset of 0 outputs the reading in degrees C. A multiplier of 1.8 and an offset of 32 converts the reading to degrees F.

In this example, the sensor is measured once a minute, and the day, time, and average temperature are output every hour. Once a day the day, time, maximum and minimum temperatures and the times they occur will be output.

Final Storage data will be sent to Storage Module. Remember, all on-line data output to a peripheral device is accomplished with Instruction 96 (Sections 4.1 and 12).

The first example described program entry one keystroke at a time. This example does not show the "A" key. Remember, "A" is used to enter and/or advance (i.e., between each line in the example below). This format is similar to the format used in EDLOG.

It's a good idea to have both the manual and the Prompt Sheet handy when going through this example. You can find the program instructions and parameters on the Prompt Sheet and can read their complete definitions in the manual.

To obtain daily output, the If Time instruction is again used to set the Output Flag and is followed by the Output Instructions to store time and the daily maximum and minimum temperatures and the time each occurs.

Any Program Control Instruction which is used to set the Output Flag high will set it low if the conditions are not met for setting it high. Instruction 92 above sets the Output Flag high every hour. The Output Instructions which follow do not output every hour because they are preceded by another Instruction 92 which sets the Output Flag high at midnight (and sets it low at any other time). This is a unique feature of Flag 0. The Output Flag is set low at the start of each table (Section 3.7).

OV5.3 EDITING AN EXISTING PROGRAM

When editing an existing program in the CR10, entering a new instruction inserts the instruction; entering a new parameter replaces the previous value.

To insert an instruction, enter the program table and advance to the position where the instruction is to be inserted (i.e., P in the data portion of the display) key in the instruction number, and then key A. The new instruction will be inserted at that point in the table, advance through and enter the parameters. The instruction that was at that point and all instructions following it will be pushed down to follow the inserted instruction.

An instruction is deleted by advancing to the instruction number (P in display) and keying #D (Table 4.2-1).

To change the value entered for a parameter, advance to the parameter and key in the correct value then press A. Note that the new value is not entered until A is keyed.

SAMPLE PROGRAM 2

Instruction # (Loc:Entry)	Parameter (Par#:Entry)	Description
*1		Enter Program Table 1
01:60		60 second (1 minute) execution interval
Key "#D" until is displayed	01:P00	Erase previous Program before continuing.
01:P17		Measure internal temperature
	01:1	Store temp in Location 1
02:P14 (differential)		Measure thermocouple temperature
	01:1	1 repetition
	02:1	Range code (2.5 mV, slow)
	03:5	Input channel of TC
	04:1	TC type: copper-constantan
	05:1	Reference temp is stored in Location 1
	06:2	Store TC temp in Location 2
	07:1	Multiplier of 1
	08:0	No offset

Instruction # (Loc.:Entry)	Parameter Par.#:Entry)	Description
03:P92		If Time instruction
	01:0	0 minutes into the interval
	02:60	60 minute interval
	03:10	Set Output Flag 0

The CR10 is programmed to measure the thermocouple temperature every sixty seconds. The If Time instruction sets the Output Flag at the beginning of every hour. Next, the Output Instructions for time and average are added.

04:P77		Output Time instruction
	01:110	Store Julian day, hour, and minute
05:P71		Average instruction
	01:1	one repetition
	02:2	Location 2 - source of TC temps. to be averaged

Instruction # (Loc.:Entry)	Parameter (Par.#:Entry)	Description
06:P92		If Time instruction
	01:0	0 minutes into the interval
	02:1440	1440 minute interval (24 hrs.)
	03:10	Set Output Flag 0
07: P77		Output Time instruction
	01:100	Store Julian day
08: P73		Maximize instruction
	01:1	One repetition
	02:10	Output time of daily maximum in hours and minutes
	03:2	Data source is Input Storage Location 2.

Instruction # (Loc.:Entry)	Parameter (Par.#:Entry)	Description
09: P74		Minimize instruction
	01:1	One repetition
	02:10	Output the time of the daily minimum in hours and minutes
	03:2	Data source is Input Storage Location 2.

The program to make the measurements and to send the desired data to Final Storage has been entered. At this point, Instruction 96 is entered to enable data transfer from Final Storage to Storage Module.

10:P96		Activate Serial Data Output.
	1:71	Output Final Storage data to Storage Module.

The program is complete. The clock must now be set so that the date and time tags are correct. (Here the example reverts back to the key by key format.)

Key	Display	Explanation
*5	00:21:32	Enter *5 Mode. Clock running but not set correctly.
A	05:00	Advance to location for year.
86	05:86	Key in year (1986).
A	05:0000	Enter and advance to location for Julian day.
197	05:197	Key in Julian day.
A	05:0021	Enter and advance to location for hours and minutes (24 hr. time).
1324	05:1324	Key in hrs.:min. (1:24 PM in this example).
A	:13:24:01	Clock set and running.
*0	LOG 1	Exit *5, compile Table 1, commence logging data.

OV6. DATA RETRIEVAL OPTIONS

There are several options for data storage and retrieval. These options are covered in detail in Sections 2, 4, and 5. Figure OV6.1-1 summarizes the various possible methods.

Regardless of the method used, there are three general approaches to retrieving data from a datalogger.

1) On-line output of Final Storage data to a peripheral storage device. On a regular schedule, that storage device is either "milked" of its data or is brought back to the office/lab where the data is transferred to the computer. In the latter case, a "fresh" storage device is usually left in the field when the full one is taken so that data collection can continue uninterrupted.

2) Bring a storage device to the datalogger and milk all the data that has accumulated in Final Storage since the last visit.

3) Retrieve the data over some form of telecommunications link, whether it be RF, telephone, short haul modem, or satellite. This can be performed under program control or by regularly scheduled polling of the dataloggers. Campbell Scientific's TELCOM program automates this process for IBM PC/XT/AT/PS-2's and compatibles.

Regardless of which method is used, the retrieval of data from the datalogger does NOT erase those data from Final Storage. The data remain in the ring memory until:

They are written over by new data (Section 2.1)

Memory is reallocated (Section 1.5)

The power to the datalogger is turned off.

Table OV6.1-1 lists the instructions used with the various methods of data retrieval.

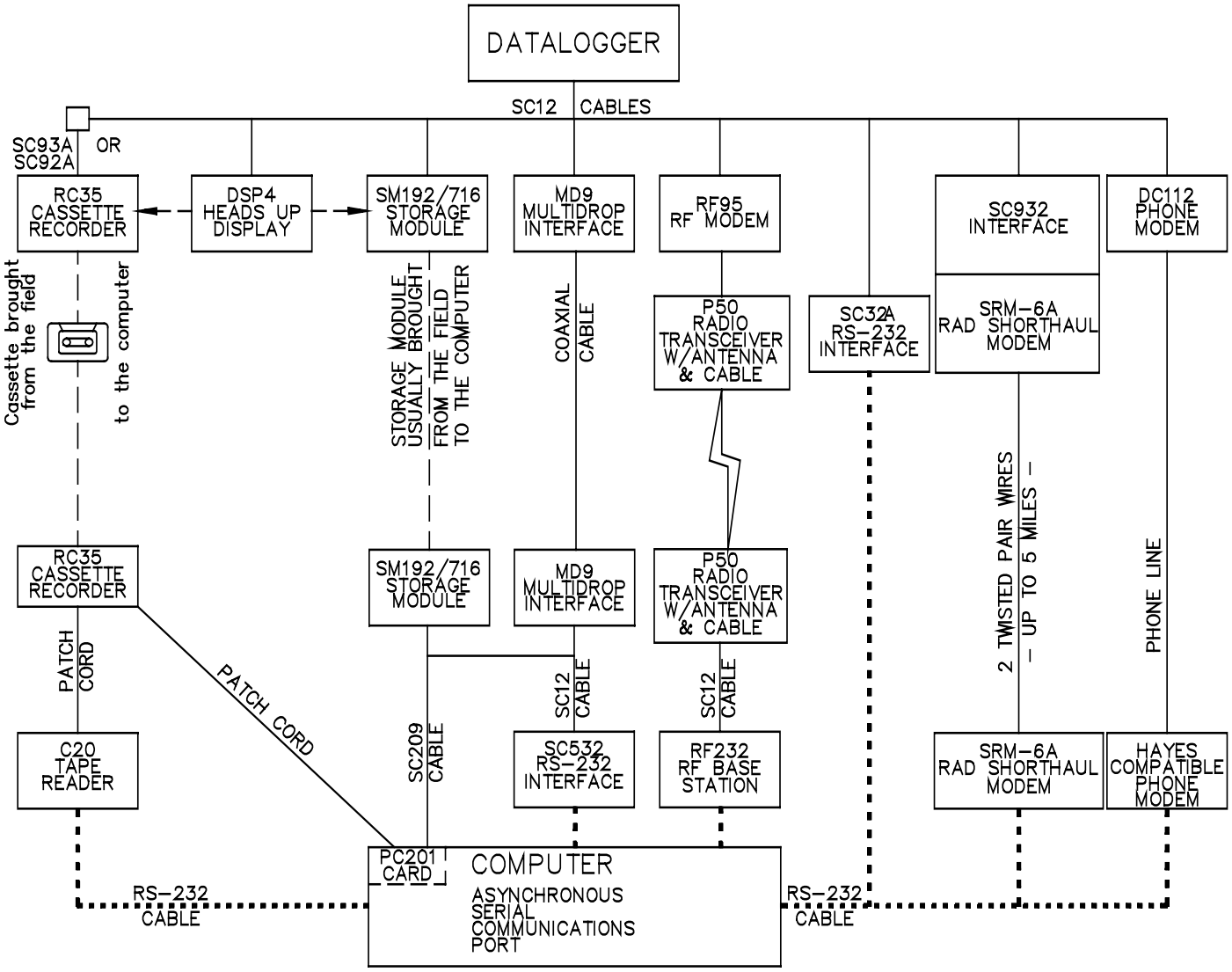
TABLE OV6.1-1. Data Retrieval Methods and Related Instructions

Storage Module	Printer, other Serial Device	Telecommunications (RF, Phone, Short Haul, SC32A)
Inst. 96, *8 *9	Inst. 96, *8 Inst. 98,	Inst. 97 (Telecommunications Commands)

TABLE OV6.1-2. Data Retrieval Sections in Manual

Instruction or Mode	Section in Manual
96	4.1, 12
Instr. 97	12
*8	4.2
*9	4.5
Telecommunications	5

DATA RETRIEVAL



NOTES:

1. ADDITIONAL METHODS OF DATA RETREIVAL ARE:
   A. SATELLITE TRANSMISSION
   B. DIRECT DUMP TO PRINTER

2. THE DSP4 HEADS UP DISPLAY ALLOWS THE USER TO VIEW DATA IN INPUT STORAGE. ALSO BUFFERS FINAL STORAGE DATA AND WRITES IT TO CASSETTE TAPE, PRINTER OR STORAGE MODULE.

3. ALL CAMPBELL SCIENTIFIC RS-232 INTERFACES HAVE A FEMALE 25 PIN RS-232 CONNECTOR.

FIGURE OV6.1-1. Data Retrieval Hardware Options

OV7. SPECIFICATIONS

The following electrical specifications are valid for an ambient temperature range of -25° to +50°C unless otherwise specified.

PROGRAM EXECUTION RATE

System tasks initiated in sync with real-time up to 64 Hz. One measurement with data transfer is possible at this rate without interruption. A single input may be measured over short intervals at rates up to 750 Hz using Burst Measurement.

ANALOG INPUTS

NUMBER OF CHANNELS: 6 differential or up to 12 single-ended. Each differential channel can be configured as two single-ended channels.

CHANNEL EXPANSION: The Model AM416 Relay Multiplexer allows an additional 64 single-ended channels to multiplex into four CR10 single-ended channels. Up to three AM416's can be connected to one CR10.

ACCURACY OF VOLTAGE MEASUREMENTS AND ANALOG OUTPUT VOLTAGES:
0.2% of FSR, 0.1% of FSR (0 to 40°C)

RANGE AND RESOLUTION: Ranges are software selectable for any channel. Resolution for a single-ended measurement is twice the value shown.

Full Scale Range	Resolution
± 2500 millivolts	333 microvolts
± 250 millivolts	33.3 microvolts
± 25 millivolts	3.33 microvolts
± 7.5 millivolts	1.00 microvolts
± 2.5 millivolts	0.33 microvolts

INPUT SAMPLE RATES: The fast A/D conversion uses a 0.25 ms signal integration time and the slow conversion uses a 2.72 ms signal integration. Two integrations, separated in time by 1/2 of an AC line cycle, are used with the 60 Hz or 50 Hz noise rejection option. Differential measurements include a second sampling with reversed input polarity to reduce thermal offset and common mode errors. Input sample rates are the time required to measure and convert the result to engineering units.

Fast single-ended voltage:	2.6 ms
Fast differential voltage:	4.2 ms
Slow single-ended voltage:	5.1 ms
Slow differential voltage:	9.2 ms
Differential with 60 Hz rejection:	25.9 ms
Fast differential thermocouple:	8.6 ms

INPUT NOISE VOLTAGE:

Fast differential	0.82 microvolts RMS
Slow differential	0.25 microvolts RMS
Differential with 60 Hz rejection	0.18 microvolts RMS

COMMON MODE RANGE: ± 2.5 volts.

DC COMMON MODE REJECTION: > 140 dB.

NORMAL MODE REJECTION: 70 dB
(60 Hz with slow differential measurement).

INPUT CURRENT: 3 nanoamps maximum.

INPUT RESISTANCE: 200 gigohms.

EXCITATION OUTPUTS

DESCRIPTION: The CR10 has 3 switched excitations, active only during measurement, with only one output active at any time. The off state is high impedance.

RANGE: ± 2.5 volts.

RESOLUTION: 0.67 millivolts.

ACCURACY: Same as voltage input.

OUTPUT CURRENT: 20 mA @ ± 2.5 V; 35 mA @ ± 2.0 V; 50 mA @ ± 1.5 V.

FREQUENCY SWEEP FUNCTION: A swept frequency, square wave output between 0 and 2.5 volts is provided for vibrating wire transducers. Timing and frequency range are specified by the instruction.

RESISTANCE AND CONDUCTIVITY MEASUREMENTS

ACCURACY: 0.015% of full scale bridge output, limited by the matching bridge resistors. The excitation voltage should be programmed so the bridge output matches the full scale input voltage range.

MEASUREMENT TYPES: 6-wire and 4-wire full bridge, 4-wire, 3-wire, and 2-wire half bridge. Bridge measurements are ratiometric and dual polarity to eliminate thermal emf. AC resistance measurements use a dual polarity 0.75 ms excitation pulse for ionic depolarization, with the signal integration occurring over the last 0.25 ms.

PERIOD AVERAGING MEASUREMENTS

DEFINITION: The time period for a specified number of cycles of an input frequency is measured, then divided by the number of cycles to obtain the average period of a single cycle.

INPUTS: Any single-ended analog channel; signal dividing or AC coupling is normally required.

INPUT FREQUENCY RANGE:

Range Code	Peak to Peak Volts Required @ Max. Freq.*	Maximum Frequency
1	2 mV	8 kHz
2	3 mV	20 kHz
3	12 mV	50 kHz
4	2000 mV	200 kHz

*AC voltage; must be centered around CR10 ground

REFERENCE ACCURACY: (-25° to 0°C) ± 80 ppm
(0° to +50°C) ± 30 ppm

RESOLUTION: ± 100 nanoseconds divided by the number of cycles measured. Resolution is reduced by signal noise and for signals with a slow transition through the zero voltage threshold.

TIME REQUIRED FOR MEASUREMENT: Signal period times the number of cycles measured plus 1.5 cycles.

PULSE COUNTERS

NUMBER OF PULSE COUNTER CHANNELS: 2 eight bit or 1 sixteen bit; software selectable.

MAXIMUM COUNT RATE: 2000 Hz, eight bit counter; 250 kHz, sixteen bit counter. Pulse counter channels are scanned at 8 Hz.

MODES: Switch closure, high frequency pulse, and low level AC.

SWITCH CLOSURE MODE
Minimum Switch Closed Time: 5 milliseconds.
Minimum Switch Open Time: 6 milliseconds.
Maximum Bounce Time: 1 millisecond open without being counted.

HIGH FREQUENCY PULSE MODE
Minimum Pulse Width: 0.002 milliseconds.
Maximum Input Frequency: 250 kHz.
Voltage Thresholds: Count upon transition from below 1.5 V to above 3.5 V.
Maximum Input Voltage: ± 20 V.

LOW LEVEL AC MODE
(Typical of magnetic pulse flow transducers or other low voltage, sine wave outputs).
Minimum AC Input Voltage: 6 mV RMS.
Input Hysteresis: 11 mV.
Maximum AC Input Voltage: 20 V RMS.

Frequency Range:

AC Input (RMS)	Range
20 mV	1 Hz to 100 Hz
50 mV	0.5 Hz to 400 Hz
150 mV to 20 V	0.3 Hz to 1000 Hz

(Consult factory if higher frequencies are desired.)

DIGITAL I/O PORTS

8 ports, software selectable as binary inputs or control outputs.

OUTPUT VOLTAGES (no load): high 5.0 V ± 0.1V; low < 0.1 V.

OUTPUT RESISTANCE: 500 Ω.

INPUT STATE: high 3.0 V to 5.5 V; low -0.5 V to 0.8 V.

INPUT RESISTANCE: 100 kΩ.

SDI-12 INTERFACE STANDARD

This communication protocol, developed for microprocessor-based hydrologic and environmental sensors, is available as a software option in the CR10.

SENSOR CONNECTIONS: Digital I/O Port #8 (for asynchronous communication), 12V power, and ground. Up to ten SDI-12 sensors can be connected to a CR10.

TRANSIENT PROTECTION

All input and output connections to the CR10 module are protected using RC filters or transzorbs connected to a heavy copper bar between the circuit card and the case. The CR10WP Wiring Panel includes additional spark gap and transzorb protection.

CPU AND INTERFACE

PROCESSOR: Hitachi 6303.

MEMORY: 32K ROM, 64K RAM.

DISPLAY: 8 digit LCD (0.5" digits).

PERIPHERAL INTERFACE: 9 pin D-type connector for keyboard display, storage module, modem, printer, card storage module, and RS-232 adapter. Baud rates selectable at 300, 1200, 9600 and 76,800. ASCII communication protocol is one start bit, one stop bit, eight data bits (no parity).

CLOCK ACCURACY: ± 1 minute per month.

SYSTEM POWER REQUIREMENTS

VOLTAGE: 9.6 to 16 volts.

TYPICAL CURRENT DRAIN: 0.7 mA quiescent, 13 mA during processing, and 35 mA during analog measurement.

BATTERIES: Any 12 volt battery can be connected as a primary power source. Several power supply options are available from Campbell Scientific.

PHYSICAL SPECIFICATIONS

SIZE: 7.8" x 3.5" x 1.5" - Measurement & Control Module; 9" x 3.5" x 2.9" - with CR10WP Wiring Panel. Additional room required for connectors.

WEIGHT: 2 lbs.

WARRANTY

Three years against defects in materials and workmanship.

SECTION 1. FUNCTIONAL MODES

1.1 PROGRAM TABLES - *1, *2, AND *3 MODES

Data acquisition and processing functions are controlled by user-entered instructions contained in program tables. Programming can be separated into 2 tables, each having its own user-entered execution interval. A third table is available for programming subroutines which may be called by instructions in Tables 1 or 2 or by a special interrupt. The *1 and *2 Modes are used to access Tables 1 and 2. The *3 Mode is used to access Subroutine Table 3.

When a program table is first entered, the display shows the table number in the ID field and 00 in the data field. Keying an "A" will advance the editor to the execution interval. If there is an existing program in the table, keying an instruction location number prior to "A" will advance directly to the instruction (e.g., 5 will advance to the fifth instruction in the table).

1.1.1 EXECUTION INTERVAL

The execution interval is entered in units of seconds as follows:

1/64 ...1 seconds, in multiples of 1/64 (0.015625)

1 .....31.875 seconds, in multiples of 1/8 (0.125)

32 .....8191 seconds, in multiples of 1 second

Execution of the table is repeated at the rate determined by this entry. The table will not be executed if 0 is entered. Entries less than 32 seconds will be rounded to a valid interval if they are within 1/512 (0.00195) second of a valid interval, otherwise error E41 will be displayed. Entries greater than 32 seconds are rounded to the nearest second.

The sample rate for a CR10 measurement is the rate at which the measurement instruction can be executed (i.e., the measurement made, scaled with the instruction's multiplier and offset, and the result placed in Input Storage). Additional processing requires extra time. The throughput rate is the rate at which a measurement can be made and the resulting value stored in Final Storage. The maximum throughput rate for fast single-ended measurements with standard software is 192 measurements per second (12 measurements repeated 16 times per second).

If the specified execution interval for a table is less than the time required to process that table, the CR10 finishes processing the table and waits for the next occurrence of the execution interval before again initiating the table (i.e., when the execution interval has elapsed and the table is still executing, that execution is skipped). Since no advantage is gained in the rate of execution with this situation, it should be avoided by specifying an execution interval adequate for the table processing time.

> **NOTE:** Whenever the processing time of the user's program exceeds a table's execution interval, an error is logged in memory. The number of overrun errors can be displayed and reset in the *B mode (Section 1.6) or using the Telecommunications A command (Section 5.1). An overrun will also cause decimal points to appear on both sides of the sixth digit of the CR10KD. The decimal points will not appear around the G in LOG if the *0 Mode is entered before the overrun occurs.

In some cases, the processing time may exceed the execution interval only when the Output Flag is set and extra time is consumed by final Output Processing. This may be acceptable. For example, suppose it is desired to sample some phenomena every 0.125 seconds and output processed data every 10 minutes. The processing time of the table which does this is less than 0.125 seconds except when output occurs (every 10 minutes). With final output the processing time is 1 second. With the execution interval set at 0.125 seconds, and a one second lag between samples once every 10 minutes, 8 measurements out of 4800 (.17%) are missed: an acceptable statistical error for most populations.

1.1.2 SUBROUTINES

Table 3 is used to enter subroutines which may be called with Program Control Instructions in Tables 1 and 2 or other subroutines. The group of instructions which form a subroutine starts with Instruction 85, Label Subroutine, and ends with Instruction 95, End (Section 12).

Subroutines 97 and 98 have the unique capability of being executed when a port goes high (ports 7 and 8 respectively). Either subroutine will interrupt Tables 1 and 2 (Section 1.1.3) when the appropriate port goes high. Port 7 cannot wake the processor, subroutine 97 will be executed at the next 1/8 second interval after the port goes high. Port 8 will wake the processor within a few microseconds. The port triggers on the rising edge (i.e., when it goes from low to high). If the port stays high the subroutine is not called again.

1.1.3 TABLE PRIORITY/INTERRUPTS

Table 1 execution has priority over Table 2. If Table 2 is being executed when it is time to execute Table 1, Table 2 will be interrupted. After Table 1 processing is completed, Table 2 processing resumes at the interruption point. If the execution interval of Table 2 coincides with Table 1, Table 1 is executed first, then Table 2.

Interrupts by Table 1 are not allowed in the middle of an instruction or while output to Final Storage is in process (flag 0 is set high). The interrupt occurs as soon as the instruction is completed or flag 0 is set low.

Special subroutines 97 and 98, initiated by a port going high (Section 1.1.2), can interrupt either Table 1 or 2 or can occur when neither is being executed. These subroutines can interrupt a table while the Output Flag is set. When the port activating 97 or 98 goes high during the execution of a table, the instruction being executed is completed before the subroutine is run (i.e., as if the subroutine was called by the next instruction).

The priority is 98, 97, Table 1, Table 2. If both 97 and 98 are pending (ports go high at the same time or both go high during the execution of the same instruction in one of the tables), 98 will be executed first. If 97 or 98 has not interrupted a table then neither table can interrupt it. 97 and 98 cannot interrupt each other. However, when 97 or 98 interrupts a table, it is as if the subroutine were in the table (e.g., if 98 interrupts Table 2, either Table 1 or 97 can interrupt it).

While 97 or 98 is being executed as a result of the respective port going high, that port interrupt is disabled (i.e., the subroutine must be completed before the port going high will have any effect).

1.1.4 COMPILING A PROGRAM

When a program is first entered, or if any changes are made in the *1, *2, *3, *A, or *C Modes, the program must be compiled before it starts running. The compile function checks for programming errors and optimizes program information for use during program execution. If errors are detected, the appropriate error codes are indicated on the display (Section 8.10). The compile function is executed when the *0 , *6, or *B Modes are entered and prior to saving a program listing in the *D Mode. The compile function is only executed after a program change has been made and any subsequent use of any of these modes will return to the mode without recompiling.

When the *0, *B, or *D Mode is used to compile, all output ports and flags are set low, the timer is reset, and data values contained in Input and Intermediate Storage are reset to zero.

When the *6 Mode is used to compile data values contained in Input Storage, the state of flags, control ports, and the timer (Instruction 26) are unaltered. Compiling always zeros Intermediate Storage.

1.2 SETTING AND DISPLAYING THE CLOCK - *5 MODE

The *5 Mode is used to display time or change the year, day or time. When "*5" is entered, the time is displayed and updated approximately once a second or longer depending on the rate and degree of data collection and processing taking place. The sequence of time parameters displayed in the *5 Mode is given in Table 1.2-1.

To set the year, day or time, enter the *5 Mode and advance to display the appropriate value. Key in the desired number and enter the value by keying "A". When a new value for hours and minutes is entered, the seconds are set to zero and current time is again displayed. To exit the *5 Mode, key "*" and the mode you wish to enter.

When the time is changed, a partial recompile is done automatically to synchronize the program with real time.

Changing time will also affect the output and execution intervals during which time is changed. Because time can only be set with a 1 second resolution, execution intervals of 1

second or less remain constant while time is reset. Averaged values will still be accurate, though the interval may have a different number of samples than normal. Totalized values will reflect the different number of samples. The pulse count instruction will use the previous interval's value if an option has been selected to discard odd intervals, otherwise it will use the count accumulated in the interval.

TABLE 1.2-1. Sequence of Time Parameters in *5 Mode

Key	Display ID:DATA	Description
*5	:HH:MM:SS	Display current time
A	05:XX	Display/enter year
A	05:XXXX	Display/enter day of year 1-365(366)
A	05:HH:MM:	Display/enter hours:minutes

1.3 DISPLAYING/ALTERING INPUT MEMORY, FLAGS, AND PORTS - *6 MODE

The *6 Mode is used to display and/or change Input Storage values and to toggle and display user flags and ports. If the *6 Mode is entered immediately following any new entries or changes in program tables, the compile function will be executed and program execution will begin.

> **NOTE:** Data values contained in Input Storage and the state of flags, control ports, and the timer (Instruction 26) are UNALTERED whenever program tables are altered and recompiled with the *6 Mode. Compiling always zeros Intermediate Storage.

TABLE 1.3-1. *6 Mode Commands

Key	Action
A	Advance to next input location or enter new value
B	Back-up to previous location
C	Change value in input location (followed by keyed in value, then "A")
D	Display/alter user flags
O	Display/alter ports
#	Display current location and allow a location number to be keyed in, followed by "A" to jump to that location

1.3.1 DISPLAYING AND ALTERING INPUT STORAGE

When "*6" is entered, the keyboard/display will read "06:0000". One can advance to view the value stored in input location 1 by keying "A". To go directly to a specific location, key in the location number before keying "A". For example, to view the value contained in Input Storage location 20, key in "*6 20 A". The ID portion of the display shows the last 2 digits of the location number. If the value stored in the location being monitored is the result of a program instruction, the value on the keyboard/display will be the result of the most recent scan and will be updated each time the instruction is executed. When using the *6 Mode from a remote terminal, a number (any number) must be sent before the value shown will be updated.

Input locations can be used to store parameters for use in computations. A value may be stored in a location, or the current value changed by keying "C" while monitoring the location, followed by the desired number and "A".

If an algorithm requires parameters to be manually modified during execution of the Program without interruption of the Table execution process, the *6 Mode can be used to change parameters stored in input locations. (If parameters will not need modification, it is better to load them from the program using Instruction 30.) If initial parameter values are required to be in place before program execution commences, use Instruction 91 at the beginning of the program table to prevent the execution until a flag is set (see the next section). Initial parameter values can be entered into input locations using the *6 Mode C command. The flag can then be set to enable the table(s).

If any program tables (*1, *2, *3) are altered and compiled in the *0 Mode after values have been entered into input locations through the *6C function, all values entered via the *6C will be set to zero. To preserve *6C entered values, always compile in the *6 Mode after altering the programming tables.

1.3.2 DISPLAYING AND TOGGLING USER FLAGS

If D is keyed while the CR10 is displaying a location value, the current status of the user flags will be displayed in the following format: "00:010010". The characters represent the flags, the left-most digit is Flag 1 and right most is Flag 8. A "0" indicates the flag is clear and a "1" indicates the flag is set. In the above example, Flags 4 and 7 are set. To toggle a flag, simply press the corresponding number. To return to displaying the input location, press "A".

Entering appropriate flag tests into the program allows manual control of program execution. For example, to manually start the execution of Table 2: enter Instruction 91 as the first instruction in Table 2. The first parameter is 25 (do if Flag 5 is low), the second parameter is 0, go to end of program table. If Flag 5 is low, all subsequent instructions in Table 2 will be skipped. Flag 5 can be toggled from the *6 Mode, effectively starting and stopping the execution of Table 2.

1.3.3 DISPLAYING AND TOGGLING PORTS

The current status of the user's ports can be displayed by hitting "0" while looking at an input location (e.g., *6A0). Ports are displayed left to right as C8, C7, ... , C1 (exactly opposite to the flags). A port configured as output can be toggled by hitting its number while in the port display mode. There is no effect on ports configured as inputs.

On power up all ports are configured as inputs. Instruction 20 is used to configure a port as an output. Ports are also configured as outputs by any program control commands which uses the port as an output (pulse, set high, set low, toggle).

1.4 COMPILING AND LOGGING DATA - *0 MODE

When the *0 Mode is entered after programming the CR10, a program compile function is executed and the display shows "LOG" followed by the program table numbers that were enabled at compilation time. The display is not updated after entering *0.

> **NOTE:** All output ports are set low, the timer is reset, and data values in Input and Intermediate Storage are RESET TO ZERO whenever the program tables are altered and the Program is recompiled with the *0 Mode. The same is true when the programs are compiled with *B or *D.

To minimize current drain, the CR10 should be left in the *0 Mode when logging data.

1.5 MEMORY ALLOCATION - *A

1.5.1 INTERNAL MEMORY

There are 2 sockets for Random Access Memory (RAM) and 1 socket which is used for (Programmable) Read Only Memory (PROM). The standard CR10 has 64K of RAM: a 32K RAM chip in each socket. Earlier versions had an 8K RAM chip in each socket. Appendix G describes how to change RAM and PROM chips.

When powered up with the keyboard display attached, the CR10KD displays HELLO while performing a self check. The total system (RAM and ROM) memory is then displayed in K bytes. The size of RAM can be displayed in the *A mode.

There are 1986 bytes allotted to Program memory. This memory may be used for 1 table or shared among all tables. Tables 3.9-1 to 3.9-4 list the amount of memory used by program instructions.

Input Storage is used to store the results of Input/Output and Processing Instructions. The values stored in input locations may be displayed using the *6 Mode (Section 1.3).

The results of Output Instructions (data used for a permanent record) are stored in Final Storage when the Output Flag is set (Section 3.7). The data in Final Storage can be monitored using the *7 Mode (Section 2.3).

Intermediate Storage is a scratch pad for Output Processing Instructions. It is used to store the results of intermediate calculations necessary for averages, standard deviations, histograms, etc. Intermediate Storage is not accessible by the user.

Each Input or Intermediate Storage location requires 4 bytes of memory. Each Final Storage location requires 2 bytes of memory. Low resolution data points require 1 Final Storage location and high resolution data points

require 2. Section 2 describes Final Storage and data retrieval in detail.

Table 1.5-1 lists the basic memory functions and the amount of memory allotted to them.

TABLE 1.5-1. Memory Allocation in CR10 (32K ROM, 64K RAM)

DEFAULT ALLOCATION

	Program Memory	System Memory	Input Storage	Intermediate Storage	Final Storage Area 1	Final Storage Area 2
64K RAM						
Bytes	1986	3302	112	256	59,816	0
Loc.			28	64	29,908	0

MAXIMUM REALLOCATION FROM FINAL STORAGE

Maximum No. of Input + Intermediate Storage Locations	Minimum No. of Final Storage Locations Area 1 + Area 2
6,862	16,368

Notes: 1) 28 is the minimum number of Input Storage locations.
2) 768 is the minimum number of Final Storage Area 1 locations.
3) 64 bytes of RAM are not used (32 in each chip).

1.5.2 *A MODE

The *A Mode is used to 1) determine the number of locations allocated to Input, Intermediate, Final Storage Area 2, and Final Storage Area 1; 2) repartition this memory; 3) check the number of bytes remaining in Program memory; 4) erase Final Storage; and 5) to completely reset the datalogger (just as if power were turned off and then on again).

A second Final Storage area (Storage Area 2) can be allocated in the *A Mode. On power up, locations allocated for Storage Area 2 defaults to 0. Final Storage Area 1 is the source from which memory is taken when Input, Intermediate, and Final Storage Area 2 memories are increased. When they are reduced, Final Storage Area 1 memory is increased. Allocation of Input and Intermediate Storage locations does NOT change Final Storage Area 2 and therefore, the data in this area are preserved.

When *A is entered, the first number displayed is the number of memory locations allocated to Input Storage. The "A" key is used to advance through the next 4 windows. Table 1.5-2 describes what the values in the *A Mode represent.

The number of memory locations allocated to Input, Intermediate and Final Storage Area 2 defaults at power-up to the values in Table 1.5-1. The size of Final Storage is determined by the size of RAM.

The sizes of Input, Intermediate and Final Storage Area 2 may be altered by keying in the desired value and entering it by keying "A". One Input or Intermediate Storage location can be exchanged for two Final Storage locations. The size of Final Storage Area 1 will be adjusted automatically.

TABLE 1.5-2. Description of *A Mode Data

Keyboard Entry	Display ID: Data	Description of Data
*A	01: XXXX	Input Storage Locations. This value can be changed by keying in the desired number (minimum of 28, maximum limited by available memory and constraints on Final Storage).
A	02: XXXX	Intermediate Storage Locations. This value can be changed by keying in the desired number (minimum of 0, maximum limited by available memory and constraints on Input and Final Storage).
A	03: XXXXX	Final Storage Area 2 Locations. Changing this number automatically reallocates Final Storage Area 1 (minimum of 0, maximum limited by available memory.)
A	04: XXXXX	Final Storage Area 1 Locations. This number is automatically altered when the number of memory locations in Input, Intermediate, or Final Storage Area 2 are changed (minimum of 768).

A	05: XXXXX	Bytes free in program memory. **Key in 1986 to completely reset datalogger.**

The maximum size of Input and Intermediate Storage and the minimum size of Final Storage are determined by the size of RAM chips installed (Table 1.5-1). Input and Intermediate Storage are confined to the same RAM chip as system and program memory, they cannot be expanded onto the second chip which is always entirely dedicated to Final Storage. A minimum 28 Input and 768 Final Storage Area 1 locations will ALWAYS be retained. The size of Intermediate Storage may be reduced to 0.

Intermediate Storage and Final Storage Area 1 are erased when memory is repartitioned. This feature may be used to clear memory without altering programming. The number of locations does not actually need to be changed; the same value can be keyed in and entered. Storage Area 2 is protected when Input and/or Intermediate Storage is reallocated, but cleared if Storage Area 2 is reallocated.

After repartitioning memory, the program must be recompiled. Compiling erases Intermediate Storage. Compiling with *0 erases Input Storage; compiling with *6 leaves Input Storage unaltered.

If Intermediate Storage size is too small to accommodate the programs or instructions entered, the "E:04" ERROR CODE will be displayed in the *0, *6, and *B Modes. The user may remove this error code by either altering the programs or by entering a larger value for Intermediate Storage size. Final Storage size can be maximized by limiting Intermediate Storage to the minimum number of locations necessary to accommodate the programs entered. The size of Final Storage and the rate at which data are stored determines how long it will take for Final Storage to fill, at which point new data will write over old.

The number of bytes remaining in program memory is displayed in the fifth window. ENTERING 1986 (the total bytes available) COMPLETELY RESETS THE CR10. All memory is erased and the power-up memory check and initialization is repeated as if the power were switched off and on again.

1.6 MEMORY TESTING AND SYSTEM STATUS - *B

The *B Mode is used to 1) read the signature of the program memory and the software PROM, 2) display the size of RAM+PROM, 3) display the number of E08 occurrences (Section 3.10), 4) display the number of overrun occurrences (Section 1.1.1), 5) display PROM version number. Table 1.6-1 describes what the values seen in the *B Mode represent. The correct signature of the CR10 PROM is listed in Appendix B.

A signature is a number which is a function of the data and the sequence of data in memory. It is derived using an algorithm which assures a 99.998% probability that if either the data or its sequence changes, the signature changes. The signature of the program memory is used to determine if the program tables have been altered. During the self check on power-up, the signature computed for a PROM is compared with a signature stored in the PROM to determine if a failure has occurred. The algorithm used to calculate the signature is described in Appendix C.

The contents of windows 6 and 7, PROM version and version revision, are helpful in determining what PROM is in the datalogger. Over the years, several different PROM versions have been released, each with operational differences. When calling Campbell Scientific for datalogger assistance, please have these two numbers available.

TABLE 1.6-1. Description of *B Mode Data

Keyboard Entry	Display ID: Data	Description of Data
*B	01: XXXXX	Program memory Signature. The value is dependent upon the programming entered and memory allotment. If the Tables have not been previously compiled, they will be compiled and run.
A	02: XXXXX	PROM Signature
A	03: XXXXX	Memory Size 32K ROM + No. K RAM
A	04: XXXXX	No. of E08 occurrences (Key in 88 to reset)
A	05: XXXXX	No. of overrun occurrences (Key in 88 to reset)
A	06: X.XXXX	PROM version number

A	07: XXXX.	Version revision number

TABLE 1.7-1. *C Mode Entries

SECURITY DISABLED

Keyboard Entry	Display ID: Data	Description
*C	01:XXXX	Non-zero password blocks entry to *1, *2, *3, *A, and *D Modes.
A	02:XXXX	Non-zero password blocks *5 and *6 except for display.
A	03:XXXX	Non-zero password blocks *5, *6, *7, *8, *9, *B, and all telecommunications commands except A, L, N, and E.

SECURITY ENABLED

Keyboard Entry	Display ID: Data	Description
*C	12:0000	Enter password. If correct, security is temporarily unlocked through that level.
A	01:XX	Level to which security has been disabled. 0 -- Password 1 entered (everything unlocked) 1 -- Password 2 entered 2 -- Password 3 entered

1.7 *C MODE -- SECURITY

The *C Mode is used to block access to the user's program information and certain CR10 functions. There are 3 levels of security, each with its own 4 digit password. All passwords are set to 0000 on power-up which disables security (unless the CR10 has a custom PROM with the password built in). Setting a password to a non-zero value "locks" the functions secured at that level. The password must subsequently be entered to temporarily unlock security through that level. Passwords are stored in write protected memory and affect the program signature.

When security is disabled, *C will advance directly to the window containing the first password. A non-zero password must be entered in order to advance to the next window. Leaving a password 0, or entering 0 for the password disables that and subsequent levels of security.

Security may be temporarily disabled by entering a password in the *C Mode or using the telecommunications L command (Section 5.1). The password entered determines what operations are unlocked (e.g., entering password 2 unlocks the functions secured by passwords 2 and 3). Password 1 (everything unlocked) must be entered before any passwords can be altered.

When security is temporarily disabled in the *C Mode, entering *0 will automatically re-enable security to the level determined by the passwords entered.

The telecommunications L command temporarily changes the security level. After hanging up, security is reset.

1.8 *D MODE -- SAVE OR LOAD PROGRAM

The *D Mode is used with Storage Module, computer/printer to save or load the user's program information (the *1, *2, *3, *A, *C, and *B Modes).

GraphTerm or TERM (PC208 software) automatically make use of the *D Mode to upload and download programs from a computer.

When "*D" is keyed in, the CR10 will display "13:00". A command (Table 1.8-1) is entered by keying the command number and "A".

TABLE 1.8-1. *D Mode Commands

Command	Description

1	Send ASCII Program
2	Load ASCII Program
7N	Save/Load/Clear Program from Storage Module N

Commands 1 and 2 (when entered from the Keyboard/Display) and 7 have an additional 2 digit option parameters (7 is entered with the Storage Module address, e.g., 71). The CR10 will display the command number and prompt for the option. If the keyboard display is not being used, the CR10 will have already set the baud rate to that of the device it is communicating with and will be ready to send or receive the file as soon as command 1 or 2 is entered.

TABLE 1.8-2. ASCII and Storage Module Command Options

Command		Option Code Description
1 & 2	1x	Synchronously addressed
	4x	Hardware enabled
		x = Baud Rate Codes 0 - 300 1 - 1200 2 - 9600 3 - 76,800
7N:00		(N is Storage Module address)
	1x	Save Program x to Storage Module (x = 1-8)
	2x	Load Program x from Storage Module (x = 1-8)
	3x	Erase Program x in Storage Module (x = 1-8)

After the option code is keyed in, key "A" to execute the command. Command 2 will be aborted if no data is received within 40 seconds.

WHEN COMMAND 2 IS EXECUTED ALL DATA IN INPUT AND INTERMEDIATE STORAGE ARE ERASED.

If the CR10 program has not been compiled when the command to save a program (1, 3 or 7) is entered, it will be compiled before the program is saved. After a command is executed, "13:0000" is displayed; *D must be entered again before another command can be given.

TABLE 1.8-3. Program Load Error Codes

E 96	Storage Module not connected or wrong address
E 97	Data not encountered within 30 sec.
E 98	Uncorrectable errors detected
E 99	Wrong type of file, or Editor Error (Section 3.10)

1.8.1 PROGRAM TRANSFER WITH COMPUTER/PRINTER

This section describes commands 1 and 2.

SENDING ASCII PROGRAM INFORMATION

Program listings are sent in ASCII. At the end of the listing, the CR10 sends control E (5 hex or decimal) twice.

Table 1.8-4 is an example of the program listing sent in response to command 1 (the actual listing is in one column but is printed in two columns to save space). Note that the listing uses numbers for each mode: The numbers for *A, *B, and *C modes are 10, 11, and 12, respectively.

TABLE 1.8-4. Example Program Listing From *D Command 1

```
MODE 1
SCAN RATE 5
1:P17
1:1

2:P86
1:10

3:P70
1:1
2:1

4:P0

MODE 2
SCAN RATE 0

MODE 3
1:P0

MODE 10
1:28
2:64
3:0
4:5332
5:1971

MODE 12
1:0
2:0

MODE 11
1:6597
2:30351
3:48
4:0
5:0
^E ^E
```

LOAD PROGRAM FROM ASCII FILE

Command 2 sets up the CR10 to load a program which is input as serial ASCII data in the same form as sent in response to command 1.

A download file need not follow exactly the same format that is used when listing a program (i.e., some of the characters sent in the listing are not really used when a program is loaded). Some rules which must be followed are:

1. "**M**" must be the first character other than a carriage return, line feed, semicolon, or 7D Hex. The "**M**" serves the same function as "*" does from the keyboard. The order in which the Modes are sent in does not matter (i.e., the information for Mode 3 could be sent before that for Mode 1).
2. "**S**" is necessary prior to the **S**can Rate (execution interval).
3. The colons (**:**) are used to mark the start of actual data.
4. A semicolon (**;**) tells the CR10 to ignore the rest of the line and can be used after an entry so that a comment can be added.

There are 4 two-character control codes which may be used to verify that the CR10 receives a file correctly:

^B ^B (2hex, 2hex)--Discard current buffer and reset signature
^C ^C (3hex, 3hex)--Send signature for current buffer
^D ^D (4hex, 4hex)--Load current buffer and reset signature
^E ^E (5hex, 5hex)--Load current buffer, Exit and compile program

As a download file is received, the CR10 buffers the data in memory; the data is not loaded into the editor or compiled until the CR10 receives a command to do so. The maximum size of the buffer is 1.5K. The minimum file that could be sent is the program listing, then ^E ^E. ^C ^C tells the CR10 to send the signature (Appendix C.3) for the current buffer of data. If this signature does not match that calculated by the sending device, ^B ^B can be sent to discard the current buffer and reset the signature. If the signature is correct, ^D ^D can be sent to tell the CR10 to load the buffer into the editor and reset the signature. Once the complete file has been sent and verified, send ^E ^E to compile the program and exit the load command.

1.8.2 PROGRAM TRANSFER WITH STORAGE MODULE

The Storage Module and Keyboard/Display or Modem/Terminal must both be connected to the CR10. After keying *D, the command 7N, is entered (N is the Storage Module address 1-8, Section 3.3). Address 1 will work with any Storage Module address. The CR10 will search for the lowest address Storage Module that is connected. The command to save, load, or clear a program and the program number (Table 1.8-2) is entered. After the operation is finished "13:0000" is displayed. Error 96 indicates that the Storage Module is not connected or the wrong address was given.

The datalogger can be programmed on power-up using a Storage Module. Storage Modules can store up to eight separate programs. If a program is stored as program number 8, and if the Storage Module is connected to the datalogger I/O at power-up, program number 8 is down loaded to the datalogger.

SECTION 2. INTERNAL DATA STORAGE

2.1 FINAL STORAGE AREAS, OUTPUT ARRAYS, AND MEMORY POINTERS

Final Storage is that portion of memory where final processed data are stored. It is from Final Storage that data is transferred to your computer or external storage peripheral.

The size of Final Storage is expressed in terms of memory locations or bytes. A low resolution data point (4 decimal characters) occupies one memory location (2 bytes), whereas a high resolution data point (5 decimal characters) requires two memory locations (4 bytes). Table 1.5-1 shows the default allocation of memory locations to Input, Intermediate, and the two Final Storage areas. The *A Mode is used to reallocate memory or erase Final Storage (Section 1.5).

The default size of Final Storage is 64K bytes or 29908 low resolution memory locations. One RAM chip is dedicated to Final Storage. This chip has 32K bytes. A minimum of 32K bytes (16K memory locations) is ALWAYS retained in Final Storage.

Final Storage can be divided into two parts: Final Storage Area 1 and Final Storage Area 2.

Final Storage Area 1 is the default storage area and the only one used if the operator does not specifically allocate memory to Area 2. A minimum of 768 memory locations will ALWAYS be retained in Final Storage Area 1.

Two Final Storage Areas may be used to:

1. Output different data to different devices.
2. Separate archive data from real time display data. In other words, you can record a short time history of real time data and separately record long term, archive data.
3. Record both high speed data (fast recording interval) and slow data without having the high speed data write over the slow data.

Each Final Storage Area can be represented as ring memory (Figure 2.1-1) on which the newest data are written over the oldest data.

The Data Storage Pointer (DSP) is used to determine where to store each new data point in the Final Storage area. The DSP advances to the next available memory location after each new data point is stored.

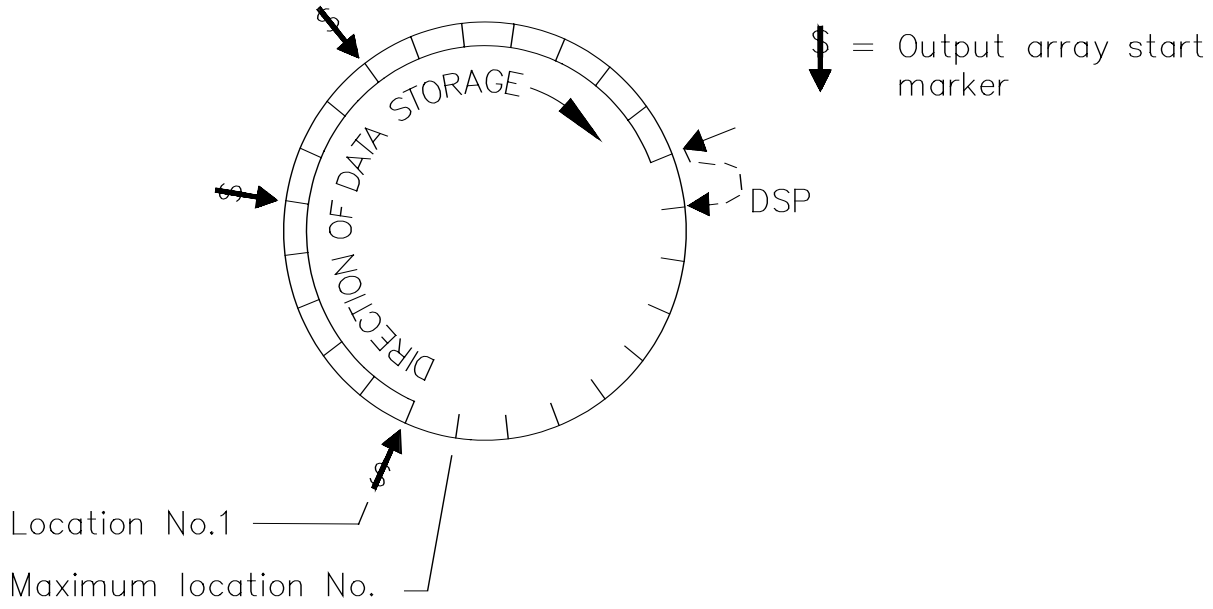


FIGURE 2.1-1. Ring Memory Representation of Final Data Storage

Output Processing Instructions store data into Final Storage only when the Output Flag is set. The string of data stored each time the Output Flag is set is called an **OUTPUT ARRAY**. The first data point in the output array is a 3 digit **OUTPUT ARRAY ID**. This ID number is set in one of two ways:

1. In the default condition, the ID consists of the program table number and the Instruction Location Number of the instruction which set the Output Flag for that particular array of data. For example, the ID of 118 in Figure 2.1-2 indicates that the 18th instruction in Table 1 set the Output Flag.
2. The output array ID can be set by the user with the second parameter of Instruction 80 (Section 11). The ID can be set to any positive integer up to 511. This option allows the user to make the output array ID independent of the programming. The program can be changed (instructions added or deleted) without changing the output array ID. This avoids confusion during data reduction, especially on long term projects where program changes or updates are likely.

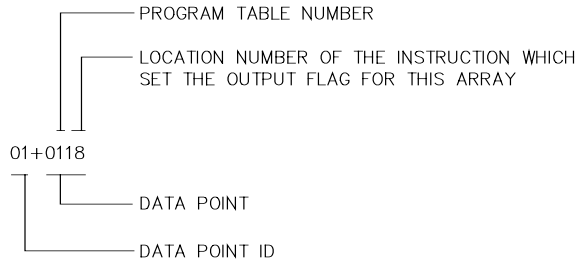

PROGRAM TABLE NUMBER
LOCATION NUMBER OF THE INSTRUCTION WHICH SET THE OUTPUT FLAG FOR THIS ARRAY
01+0118
DATA POINT
DATA POINT ID

FIGURE 2.1-2. Output Array ID

> **NOTE:** If Instruction 80 is used to designate the active Final Storage Area and parameter 2 is 0, the output array ID is determined by the position of Instruction 80 or by the position of the instruction setting the Output Flag, whichever occurs last.

A start-of-array marker ($ in Figure 2.1-1) is written into Final Storage with the Output Array ID. This marker is used as a reference point from which to number the data points of the output array. The start of array marker occupies the same Final Storage location as the Array ID and is transparent for all user operations.

Data are stored in Final Storage before being transmitted to an external device. There are 5 pointers for each Final Storage Area which are used to keep track of data transmission. These pointers are:

1. Display Pointer (**DPTR**)
2. Tape Pointer (**TPTR**)
3. Printer Pointer (**PPTR**)
4. Telecommunications (Modem) Pointer (**MPTR**)
5. Storage Module Pointer (**SPTR**)

The **DPTR** is used to recall data to the keyboard/ display. The positioning of this pointer and data recall are controlled from the keyboard (*7 Mode).

The **TPTR** is used to control data transmission to a cassette tape recorder. When on-line tape transfer is activated (Instruction 96, option 00), data is transmitted to tape whenever the **DSP** is a minimum of 512 memory locations ahead of the TPTR. The TPTR may also be positioned via the keyboard for manually initiated data transfer to tape (*8 Mode).

The **PPTR** is used to control data transmission to a printer or other serial device. Whenever on-line printer transfer is activated (Instruction 96), data between the PPTR and DSP are transmitted. The PPTR may also be positioned via the keyboard for manually initiated data transmission (*8 Mode).

The **MPTR** is used in transmitting data over a telecommunications interface. When telecommunications is first entered, the MPTR is set to the same location as the DSP. Positioning of the MPTR is then controlled by commands from the external calling device (Section 5.1).

The **SPTR** is used to control data transmission to a Storage Module. When on-line transfer is activated by Instruction 96, data is transmitted each time an output array is stored in Final Storage IF THE STORAGE MODULE IS CONNECTED TO THE CR10. If the Storage Module is not connected, the CR10 does not transmit the data nor does it advance the SPTR to the new DSP location. It saves the data until the Storage Module is connected. Then, during the next execution of Instruction 96, the CR10 outputs all of the data between the SPTR and the DSP and updates the SPTR to the DSP location (Section 4.1)

The SPTR may also be positioned via the keyboard for manually initiated data transfer to the Storage Module (*8 Mode, Section 3.2.3).

NOTE: All memory pointers are set to the DSP location when the datalogger compiles a program. For this reason, ALWAYS RETRIEVE UNCOLLECTED DATA BEFORE MAKING PROGRAM CHANGES. For example, assume the TPTR lags the DSP by less than 512 data points when the datalogger program is altered. On compiling, the TPTR is positioned with the DSP, losing reference to the data that was intended to be transferred to tape. The data is not automatically transferred and appears as a discontinuity in the data file. Until the ring memory wraps around and data overwrite occurs, the data may be recovered using the *8 Mode. This scenario is also true for the SPTR and data intended for a Storage Module.

2.2 DATA OUTPUT FORMAT AND RANGE LIMITS

Data is stored internally in Campbell Scientific's Binary Final Storage Format (Appendix C.2). Data may be sent to Final Storage in either LOW RESOLUTION or HIGH RESOLUTION format.

2.2.1 RESOLUTION AND RANGE LIMITS

Low resolution data is a 2 byte format with 4 significant digits and a maximum magnitude of +6999. High resolution data is a 4 byte format with 5 significant digits and a maximum possible output value of +99999 (see Table 2.2-1 below).

TABLE 2.2-1. Resolution Range Limits of CR10 Data

Resolution	Zero	Minimum Magnitude	Maximum Magnitude
Low	0.000	+0.001	+6999.
High	0.0000	+ .00001	+99999.

The resolution of the low resolution format is reduced to 3 significant digits when the first (left most) digit is 7 or greater. Thus, it may be necessary to use high resolution output or an offset to maintain the desired resolution of a measurement. For example, if water level is to be measured and output to the nearest 0.01 ft., the level must be less than 70 ft. for low resolution output to display the 0.01 ft. increment. If the water level was expected to range from 50 to 80 feet the data could either be output in high resolution or could be offset by 20 ft. (transforming the range to 30 to 50 ft.).

2.2.2 INPUT AND INTERMEDIATE STORAGE DATA FORMAT

While output data have the limits described above, the computations performed in the CR10 are done in floating point arithmetic. In Input and Intermediate Storage, the numbers are stored and processed in a binary format with a 23 bit binary mantissa and a 6 bit binary exponent. The largest and smallest numbers that can be stored and processed are 9×10^{18} and 1×10^{-19}, respectively. The size of the number determines the resolution of the arithmetic. A rough approximation of the resolution is that it is better than 1 in the seventh digit. For example, the resolution of 97,386,924 is better than 10. The resolution of 0.008673192 4 is better than 0.000000001.

A precise calculation of the resolution of a number may be determined by representing the number as a mantissa between .5 and 1 multiplied by 2 raised to some integer power. The resolution is the product of that power of 2 and 2^{-24}. For example, representing 478 as $.9336 * 2^9$, the resolution is $2^9 * 2^{-24} = 2^{-15} = 0.0000305$. A description of Campbell Scientific's floating point format may be found in the description of the J and K Telecommunications Commands in Appendix C.

2.3 DISPLAYING STORED DATA ON KEYBOARD/DISPLAY - *7 MODE

(Computer/terminal users refer to Section 5 for instructions on entering the Remote Keyboard State.)

Final Storage may be displayed by using the *7 Mode. Key *7.

If you have allocated memory to Final Storage Area 2, the display will show:

07:00

Select which Storage Area you wish to view:

00 or 01 = Final Storage Area 1
02 = Final Storage Area 2

If no memory has been allocated to Final Storage Area 2, this first window will be skipped.

The next window displays the current DSP location. Pressing A advances you to the Output array ID of the oldest Array in the Storage Area. To locate a specific Output Array, enter a location number that positions the Display Pointer (DPTR) behind the desired data and press the "A" key. If the location number entered is in the middle of an Output Array, the DPTR is automatically advanced to the first data point of the next Output Array. Repeated use of the "A" key advances through the Output Array, while use of the "B" key backs the DPTR through memory.

The memory location of the data point is displayed by pressing the "#" key. At this point, another memory location may be entered, followed by the "A" key to jump to the start of the Output Array equal to or just ahead of the location entered. Whenever a location number is displayed by using the "#" key, the corresponding data point can be displayed by pressing the "C" key.

The same element in the next Output Array with the same ID can be displayed by hitting #A. The same element in the previous array can be displayed by hitting #B. If the element is 1 (Array ID), then #A advances to the next array and #B backs up to the previous array. #0A backs up to the start of the current array.

The keyboard commands used in the *7 Mode are summarized in Table 2.3-1.

Advancing the DPTR past the Data Storage Pointer (DSP) displays the oldest data point. Upon entering the *7 Mode, the oldest Output Array can be accessed by pressing the "A" key.

TABLE 2.3-1. *7 Mode Command Summary

Key	Action
A	Advance to next data point
B	Back-up to previous data point
#	Display location number of currently displayed data point value
C	Display value of current location
#A	Advance to same element in next Output Array with same ID
#B	Back-up to same element in previous Output Array with same ID
#0A	Back-up to the start of the current Final Data Storage Array
*	Exit *7 Mode

SECTION 3. INSTRUCTION SET BASICS

The instructions used to program the CR10 are divided into four types: Input/Output (I/O), Processing, Output Processing, and Program Control. I/O Instructions are used to make measurements and store the readings in input locations or to initiate analog or digital port output. Processing Instructions perform mathematical operations using data from Input Storage locations and place the results back into specified Input Storage locations. Output Processing Instructions provide a method for generating time or event dependent data summaries from processed sensor readings residing in specified Input Storage locations. Program Control Instructions are used to direct program execution based on time and or conditional tests on input data and to direct output to external devices.

Instructions are identified by a number. There are a fixed number of parameters associated with each instruction to give the CR10 the information required to execute the instruction. The set of instructions available in the CR10 is determined by the PROM (Programmable Read Only Memory) inside the CR10. Appendix B lists the PROM options available.

3.1 PARAMETER DATA TYPES

There are 3 different data types used for Instruction parameters: Floating Point (FP), 4 digit integers (4), and 2 digit integers (2). The parameter data type is identified in the listings of the instruction parameters in Sections 9-12. Different data types are used to allow the CR10 to make the most efficient use of its memory.

Floating Point parameters are used to enter numeric constants for calibrations or mathematical operations. While it is only possible to enter 5 digits (magnitude +.00001 to +99999.), the internal format has a much greater range ($1x10^{-19}$ to $9x10^{18}$, Section 2.2.1). Instruction 30 can be used to enter a number in scientific notation into an input location.

3.2 REPETITIONS

The repetitions parameter on many of the I/O, Processing, and Output Processing Instructions is used to repeat the instruction on a number of sequential Input Channels or Input Storage locations. For example, if you are making 4 differential voltage measurements on the same voltage range, wire the inputs to sequential channels and enter the Differential Voltage Measurement Instruction once with 4 repetitions, rather than entering 4 separate measurement instructions. The instruction will make 4 measurements starting on the specified channel number and continuing through the 3 succeeding differential channels. The results will be stored in the specified input location and the 3 succeeding input locations. Averages for all 4 measurements can be calculated by entering the Average Instruction with 4 repetitions.

When several of the same type of measurements will be made, but the calibrations of the sensors are different, it requires less time to make the measurements using one measurement with repetitions and then apply the calibrations with a scaling array (Inst. 53) than it does to enter the instruction several times in order to use a different multiplier and offset. This is due to set up and calibration time for each measurement instruction. However, if time is not a constraint, separate instructions may make the program easier to follow.

3.3 ENTERING NEGATIVE NUMBERS

After keying in a number, press C or "-" to change the number's sign. On floating point numbers a minus sign (-) will appear to the left of the number. Excitation voltages in millivolts for I/O Instructions are 4 digit integers; when C is keyed 2 minus signs (-) will appear to the right of the number indicating a negative excitation. Even though this display is the same as that indicating an indexed input location, (Section 3.4) there is no indexing effect on excitation voltage.

3.4 INDEXING INPUT LOCATIONS AND PORTS

When used within a loop, the parameters for input locations and the commands to set, toggle, or pulse a port can be Indexed to the loop counter. The loop counter is added to the indexed value to determine the actual Input

Location or Port the instruction acts on. Normally the loop counter is incremented by 1 after each pass through the loop. Instruction 90, Step Loop Index, allows the increment step to be changed. See Instructions 87 and 90, Section 12, for more details.

To index an input location (4 digit integer) or set port command (2 digit integer) parameter, C or "-" is pressed after keying the value but before entering the parameter. Two minus signs (-) will be displayed to the right of the parameter.

3.5 VOLTAGE RANGE AND OVERRANGE DETECTION

The voltage RANGE code parameter on Input/Output Instructions is used to specify the full scale range of the measurement and the integration period for the measurement (Table 3.5-1).

The full scale range selected should be the smallest that will accommodate the full scale output of the sensor being measured. Using the smallest possible range will result in the best resolution for the measurement.

Four different integration sequences are possible. The relative immunity of the integration sequences to random noise is: 60 Hz rej. = 50 Hz rej. > 2.72ms integ. > 272 μs integ. The 60 Hz rejection integration rejects noise from 60 Hz AC line power. The 50 Hz rejection is for countries whose electric utilities operate at 50 Hz (Section 13.1).

When a voltage input exceeds the range programmed, the value which is stored is set to the maximum negative number and displayed as -99999 in high resolution or -6999 in low resolution.

An input voltage greater than +5 volts on one of the analog inputs will result in errors and possible overranging on the other analog inputs. Voltages greater than 16 volts may permanently damage the CR10.

> **NOTE**: Voltages in excess of 5.5 volts applied to a control port can cause the CR10 to malfunction.

3.6 OUTPUT PROCESSING

Most Output Processing Instructions require both an Intermediate Data Processing operation and a Final Data Processing operation. For example, when the Average Instruction, 71, is initiated, the intermediate processing operation increments a sample count and adds each new Input Storage value to a cumulative total residing in Intermediate Storage. When the Output Flag is set, the final processing operation divides the cumulative total by the number of samples to find the average. The average is then stored in final storage and the cumulative total and number of samples are set to zero in Intermediate Storage.

Final Storage Area 1 (Sections 1.5, 2.1) is the default destination of data output by Output Processing Instructions. Instruction 80 may be used to direct output to either Final Storage Area 2 or to Input Storage.

Output Processing Instructions requiring intermediate processing sample the specified input location(s) each time the Output Instruction is executed, NOT each time the location value is updated by an I/O Instruction. For example: Suppose a temperature measurement is initiated by Table 1 which has an execution interval of 1 second.

TABLE 3.5-1. Input Voltage Ranges and Codes

Range Code				Full Scale Range	Resolution*
Slow 2.72ms Integ.	Fast 250 us Integ.	60 Hz Reject.	50 Hz Reject.		
1	11	21	31	±2.5 mV	0.33 μV
2	12	22	32	±7.5 mV	1. μV
3	13	23	33	±25 mV	3.33 μV
4	14	24	34	±250 mV	33.3 μV
5	15	25	35	±2500 mV	333. μV

***** *Differential measurement, resolution for single-ended measurement is twice value shown.*

The instructions to output the average temperature every 10 minutes are in Table 2 which has an execution interval of 10 seconds. The temperature will be measured 600 times in the 10 minute period, but the average will be the result of only 60 of those measurements because the instruction to average is executed only one tenth as often as the instruction to make the measurement.

Intermediate Processing can be disabled by setting Flag 9 which prevents Intermediate Processing without actually skipping over the Output Instruction (Section 1.2).

All of the Output Processing Instructions store processed data values when and only when the Output Flag is set (Section 1.2). The Output Flag (Flag 0) is set at desired intervals or in response to certain conditions by using an appropriate Program Control Instruction (Section 11).

3.7 USE OF FLAGS: OUTPUT AND PROGRAM CONTROL

There are 10 flags which may be used in CR10 programs. Two of the flags are dedicated to specific functions: Flag 0 causes Output Processing Instructions to write to Final Storage, and Flag 9 disables intermediate processing. Flags 1-8 may be used as desired in programming the CR10. Flags 0 and 9 are automatically set low at the beginning of the program table. Flags 1-8 remain unchanged until acted on by a Program Control Instruction or until manually toggled from the *6 Mode.

TABLE 3.7-1. Flag Description

Flag 0	-	Output Flag
Flag 1 to 8	-	User Flags
Flag 9	-	Intermediate Processing Disable Flag

Flags are set with Program Control Instructions. The Output Flag (Flag 0) and the intermediate programming disable flag (Flag 9) will always be set low if the set high condition fails. The status of flags 1-8 does not change when a conditional test is false.

3.7.1 THE OUTPUT FLAG

A group of processed data values is placed in Final Data Storage by Output Processing Instructions when the Output Flag (Flag 0) is set high. This group of data is called an Output Array. The Output Flag is set using Program Control Instructions according to time or event dependent intervals specified by the user. The Output Flag is set low at the beginning of each table.

Output is most often desired at fixed intervals; this is accomplished with Instruction 92, If Time. Output is usually desired on the even interval, so Parameter 1, time into the interval, is 0. The time interval (Parameter 2), in minutes, is how often output will occur; i.e., the Output Interval. The command code (Parameter 3) is 10, causing Flag 0 to be set high. The time interval is synchronized to 24 hour time; output will occur on each integer multiple of the Output Interval starting from midnight (0 minutes). If the Output Interval is not an even divisor of 1440 minutes (24 hours), the last output interval of the day will be less than the specified time interval. Output will occur at midnight and will resume synchronized to the new day. Instruction 92 is followed in the program table by the Output Instructions which define the Output Array desired.

Each group of Output Processing Instructions creating an Output Array is preceded by a Program Control Instruction that sets the Output Flag.

NOTE: If the Output Flag is already set high and the test condition of a subsequent Program Control Instruction acting on Flag 0 fails, the flag is set low. This eliminates entering another instruction to specifically reset the Output Flag before proceeding to another group of Output Instructions with a different output interval.

3.7.2 THE INTERMEDIATE PROCESSING DISABLE FLAG

The Intermediate Processing Disable Flag (Flag 9) suspends intermediate processing when it is set high. This flag is used to restrict sampling for averages, totals, maxima, minima, etc., to times when certain criteria are met. The flag is automatically set low at the beginning of the program table.

As an example, suppose it is desired to obtain a wind speed rose incorporating only wind speeds greater than or equal to 4.5 m/s. The wind speed rose is computed using the Histogram Instruction 75, and wind speed is stored in input location 14, in m/s. Instruction 89 is placed just before Instruction 75 and is used to set Flag 9 high if the wind speed is less than 4.5 m/s:

TABLE 3.7-2. Example of the Use of Flag 9

Inst. Loc.	Param. No.	Entry	Description
X	P	89	If wind speed < 4.5 m/s
	1	14	Wind speed location
	2	4	Comparison: <
	3	4.5	Minimum wind speed for histogram
	4	19	Set Flag 9 high
X+1	P	75	Histogram
X+2	P	86	Do
	1	29	Set Flag 9 Low

NOTE: Flag 9 is automatically reset the same as Flag 0. If the intermediate processing disable flag is already set high and the test condition of a subsequent Program Control Instruction acting on Flag 9 fails, the flag is set low. This feature eliminates having to enter another instruction to specifically reset Flag 9 before proceeding to another group of test conditions.

3.7.3 USER FLAGS

Flags 1-8 are not dedicated to a specific purpose and are available to the user for general programming needs. The user flags can be manually toggled from the keyboard in the *6 Mode (Section 1.3). By inserting the flag test (Instruction 91) at appropriate points in the program, the user can use the *6 Mode to manually direct program execution.

3.8 PROGRAM CONTROL LOGICAL CONSTRUCTIONS

Most of the Program Control Instructions have a command code parameter which is used to specify the action to be taken if the condition tested in the instruction is true. Table 3.8-1 lists these codes.

TABLE 3.8-1. Command Codes

0	- Go to end of program table
1-9, 79-99	- Call Subroutine 1-9, 79-99[1]
10-19	- Set Flag 0-9 high
20-29	- Set Flag 0-9 low
30	- Then Do
31	- Exit loop if true
32	- Exit loop if false
41-48	- Set Port 1-8 high[2]
51-58	- Set Port 1-8 low[2]
61-68	- Toggle Port 1-8[2]
71-78	- Pulse Port 1-8[2]

[1] 97 and 98 are special subroutines which can be called by Control ports 7 and 8 going high; see Instruction 85 for details.

[2] Ports can be indexed to the loop counter (Section 3.4).

3.8.1 IF THEN/ELSE COMPARISONS

Program Control Instructions can be used for If then/else comparisons. When Command 30 (Then do) is used with Instructions 83 or 88-92, the If Instruction is followed immediately by instructions to execute if the comparison is true. The Else Instruction (94) is optional and is followed by the instructions to execute if the comparison is false. The End Instruction (95) ends the If then/else comparison and marks the beginning of the instructions that are executed regardless of the outcome of the comparison (see Figure 3.8-1).

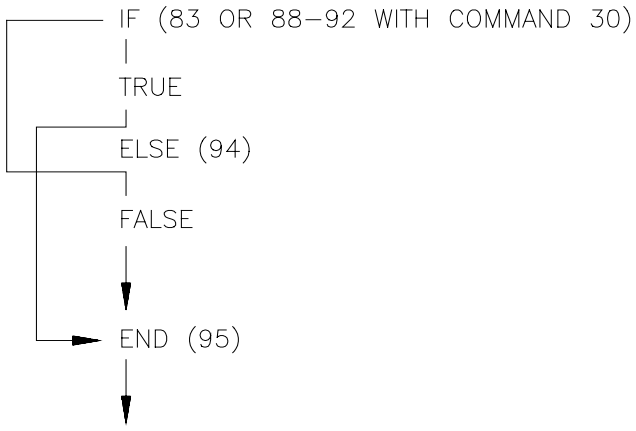


FIGURE 3.8-1. If Then/Else Execution Sequence

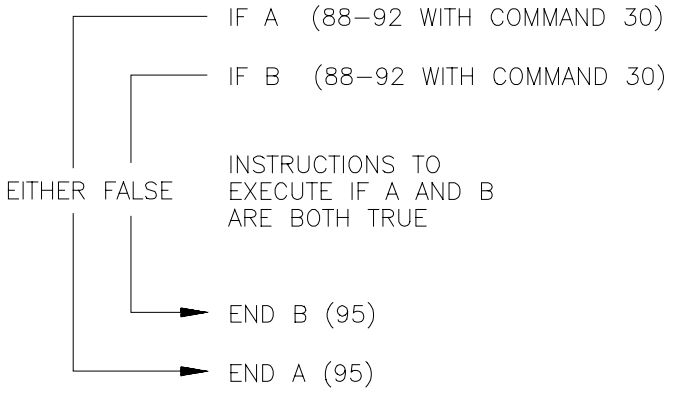


FIGURE 3.8-2. Logical AND Construction

If Then/Else comparisons may be nested to form logical AND or OR branching. Figure 3.8-2 illustrates an AND construction. If conditions A and B are true, the instructions included between IF B and the first End Instruction will be executed.

If either of the conditions is false, execution will jump to the corresponding End Instruction, skipping the instructions between.

A logical OR construction is also possible. Figure 3.8-3 illustrates the instruction sequence that will result in subroutine X being executed if either A or B is true.

```
IF A  (88-92 with command 30)
Call subroutine X (86, command=X)
ELSE  (94)
IF B  (88-92 with command 30)
Call subroutine X (86, command=X)
END B (95)
END A (95)
```

FIGURE 3.8-3. Logical OR Construction

A logical OR can also be constructed by setting a flag if a comparison is true. (The flag is cleared before making comparisons.) After all comparisons have been made, execute the desired instructions if the flag is set.

The Begin Case Instruction 93 and If Case Instruction 83 allow a series of tests on the value in an input location. The case test is started with Instruction 93 which specifies the location to test. A series of Instruction 83s are then used to compare the value in the location with fixed values. When the value in the input location is less than the fixed value specified in Instruction 83, the command in that Instruction 83 is executed, and execution branches to the END Instruction 95 which closes the case test (see Instruction 93).

3.8.2 NESTING

A branching or loop instruction which occurs before a previous branch or loop has been closed is nested. The maximum nesting level is 9 deep. Loop Instruction 87 and Begin Case Instruction 93 both count as 1 level. Instructions 83, 86, 88, 89, 91, and 92 each count as one level when used with the Command "30" which is the "Then Do" command. Use of Else, Instruction 94, also counts as one nesting level each time it is used. For example, the AND construction above is nested 2 deep while the OR construction is nested 3 deep.

Subroutine calls do not count as nesting with the above instructions, though they have their own nesting limit (7, see Instruction 85). Branching and loop nesting start at zero in each subroutine.

Any number of groups of nested instructions may be used in any of the three Programming Tables. The number of groups is only restricted by the program memory available.

3.9 INSTRUCTION MEMORY AND EXECUTION TIME

There are 1986 bytes of program memory available for the programs entered in the *1, *2, and *3 Program Tables. Each instruction also makes use of varying numbers of Input, Intermediate, and Final Storage locations. The tables list the memory used by each instruction and the approximate time required to execute it.

When attempting to make a series of measurements and calculations at a fast rate, it is important to examine the time required for the automatic calibration sequence and possibly make use of the program controlled calibration, Instruction 24. Section 13.9 describes the calibration process.

TABLE 3.9-1. Input/Output Instruction Memory and Execution Times

R = No. of Reps.

INSTRUCTION		MEMORY		EXECUTION TIME (ms) MEASUREMENT RANGE							
		INPUT LOC.	PROG. BYTES	1-4 or NA	5	11-14	15	21-24	25	31-34	35
1	VOLT (SE)	R	15	4.5 + 5.1R	2.1 + 2.7R	2.1 + 2.6R	1.9 + 2.5R	12.9 + 13.4R	10.5 + 11.1R	14.8 + 15.1R	12.1 + 12.7R
2	VOLT (DIFF)	R	15	0.5 + 9.2R	0.5 + 4.4R	0.5 + 4.2R	0.5 + 3.8R	0.5 + 25.9R	0.5 + 21.3R	0.5 + 29.4R	0.5 + 24.7R
3	PULSE	R	15	0.9 + 1.0R							
4	EX-DEL-SE	R	20	4.7 + 5.3R	2.3 + 2.8R	2.3 + 2.8R	2.1 + 2.6R	13.1 + 13.6R	10.7 + 11.2R	15.0 + 15.2R	12.3 + 12.8R
5	AC HALF BR	R	18	6.9 + 9.6R	4.4 + 4.8R	4.4 + 4.7R	4.3 + 4.4R	15.0 + 26.3R	11.3 + 21.7R	16.8 + 29.6R	14.5 + 24.8R
6	FULL BR	R	18	3.3 + 17.7R	3.3 + 8.0R	3.3 + 7.8R	3.2 + 7.0R	3.2 + 51.1R	3.3 + 41.7R	3.3 + 57.8R	3.3 + 48.5R
7	3W HALF BR	R	18	4.5 + 21.6R	2.8 + 11.8R	3.6 + 11.6R	3.4 + 10.9R	14.9 + 54.6R	10.4 + 45.3R	15.9 + 61.4R	14.3 + 51.6R
8	EX-DEL-DIFF	R	20	4.7 + 5.3R	2.3 + 2.9R	2.4 + 2.9R	2.1 + 2.7R	13.1 + 13.6R	10.6 + 11.2R	15.0 + 15.3R	12.3 + 12.9R
	above w/o delay; w/delay:			F + 2R(Tr + delay) where F is the fixed time and Tr is the time/rep from above.							
9	FULL BR-MEX	R	19	1.3 + 37.3R	1.3 + 18.0R	1.3 + 17.6R	1.3 + 15.9R	1.3 + 104.3R	1.3 + 85.0R	1.3 + 117.9R	1.3 + 98.3R
	V1 on range 5, V2 on:			1.3 + 27.8R	1.3 + 18.0R	1.3 + 17.7R	1.3 + 16.9R	1.3 + 61.2R	1.3 + 51.8R	1.3 + 67.8R	1.3 + 58.2R
	V1 on range 15, V2 on:			1.3 + 26.7R	1.3 + 16.9R	1.3 + 16.7R	1.3 + 15.9R	1.3 + 59.9R	1.3 + 50.6R	1.3 + 66.6R	1.3 + 57.4R
10	BATT. VOLT	1	4	4.8							
11	TEMP-(107)	R	15	4.5 + 7.2R							
12	RH (207)	R	17	2.1 + 9.1R							
13	TEMP-TC SE	R	18	2.4 + 10.2R	6.7 + 6.4R	4.9 + 6.6R	2.5 + 6.5R	15.7 + 17.3R	12.1 + 15.2R	17.0 + 19.2R	18.9 + 15.6R
14	TEMP-TC DIF	R	18	3.7 + 13.5R	3.7 + 8.2R	3.7 + 8.6R	3.7 + 7.7R	3.7 + 30.4R	3.7 + 25.4R	3.7 + 33.5R	3.7 + 28.1R
16	TEMP-RTD	R	15	0.4 + 3.0R							
17	TEMP-INTERNL	1	4	6.2							
18	TIME	1	7	1.7							
19	SIGNATURE	1	4	750							
20	PORT SET	0	6	9.5							
22	EXCIT-DEL	0	9	0.5							
24	CALIBRATION	19	4	2664, 21.2 when only saving results of automatic calibration.							
25	READ PORTS	1	6	0.4							
26	TIMER	1 or 0	4	0.3 to reset, 0.8 to load into location.							
27	PER.AVG.	R	19	period of signal * (No. cycles + 1.5), or time limit + 2							
28	VIB.WIRE	R	21	170 + period of signal * (No. cycles + 1.5), or time limit +172							
101	SDM-INT8	1-8	25	2.3 + 1.65 * No.Values+averaging interval if used							
102	SDM-SW8A	R	16	4 + F * R; F=function time = 1 for state, 2 for counts or cycle, 0 for signature							
103	SDM-AO4	0	6	1.7 + 0.8R							
104	SDM-CD16	0	6	2R							

TABLE 3.9-2. Processing Instruction Memory and Execution Times R = No. of Reps.

INSTRUCTION	INPUT LOC.	MEMORY INTER. LOC.	PROG. BYTES	EXECUTION TIME (ms)
30 Z=F	1	0	9	0.2 + 0.6 * exponent
31 Z=X	1	0	6	0.5
32 Z=Z+1	1	0	4	0.6
33 Z=X+Y	1	0	8	1.1
34 Z=X+F	1	0	10	0.9
35 Z=X-Y	1	0	8	1.1
36 Z=X*Y	1	0	8	1.2
37 Z=X*F	1	0	10	0.9
38 Z=X/Y	1	0	8	2.7
39 Z=SQRT(X)	1	0	6	12.0
40 Z=LN(X)	1	0	6	7.4
41 Z=EXP(X)	1	0	6	5.9
42 Z=1/X	1	0	6	2.6
43 Z=ABS(X)	1	0	6	0.7
44 Z=FRAC(X)	1	0	6	0.3
45 Z=INT(X)	1	0	6	1.0
46 Z=X MOD F	1	0	10	3.2
47 Z=XY	1	0	8	13.3
48 Z=SIN(X)	1	0	6	6.5
49 SPA. MAX	1 or 2	0	8	1.5 + 0.9 (swath-1)
50 SPA. MIN	1 or 2	0	8	1.7 + 0.9 (swath-1)
51 SPA. AVG	1	0	8	3.3 + 0.6 (swath-1)
53 A*X+B	4	0	36	4.1
54 BLOCK MOVE	R	0	10	0.2 + 0.2R
55 POLYNOMIAL	R	0	31	1.2 + (2.0 + 0.4 * order)R
56 SAT. VP	1	0	6	4.2
57 WDT-VP	1	0	10	8.1
58 LP FILTER	R	R	13	0.5 + 2.2R
59 X/(1-X)	1	0	9	0.4 + 3.0R
61 INDIR. MOVE	1	0	6	0.4 neither indexed, 0.5 1 location indexed, 0.7 both locations indexed
63 PARA.EXTN.	0	0	10	0.1
66 ARC TAN	1	0	8	6.7

TABLE 3.9-3. Output Instruction Memory and Execution Times R = No. of Reps.

INSTRUCTION	INTER. LOC.	MEM. FINAL VALUES[1]	PROG. BYTES	EXECUTION TIME (ms) FLAG 0 LOW	FLAG 0 HIGH
69 WIND VECTOR	2+9R	(2, 3, or 4)R	12		
		Options 00, 10, 20		3.5 + 17.5R	3.5 + 75R
		Options 01, 11, 21		3.5 + 16R	3.5 + 30R
70 SAMPLE	0	R	6	0.1	0.4+ 0.6R
71 AVERAGE	1+R	R	7	0.9+ 0.5R	2.1+ 3.0R
72 TOTALIZE	R	R	7	0.6+ 0.5R	1.1+ 1.0R
73 MAXIMIZE	(1 or 2)R	(1,2, or 3)R	8	0.9+ 1.7R	1.3+ 2.8R
74 MINIMIZE	(1 or 2)R	(1,2, or 3)R	6	0.9+ 1.7R	1.3+ 2.8R
75 HISTOGRAM	1+bins*R	bins*R	24	0.4+ 3.1R	0.9+ (3.3+2.8*bins)R
77 REAL TIME	0	1 to 4	4	0.1	1.0
78 RESOLUTION	0	0	3	0.4	0.4
79 SMPL ON MM	R	R	7	0.3	1.1
80 STORE AREA[1]	0	0	7	0.2	0.2
82 STD. DEV.	1+3R	R	7	1.0+ 1.4R	1.8+ 2.2R

[1]Output values may be sent to either Final Storage area or Input Storage with Instruction 80.

TABLE 3.9-4. Program Control Instruction Memory and Execution Times

INSTRUCTION	MEMORY INTER. LOC.	MEMORY PROG. BYTES	EXECUTION TIME (ms)
83 IF CASE <F	0	9	0.5
85 LABEL SUBR.	0	3	0
86 DO	0	5	0.1
87 LOOP	1	7	0.2
88 IF X<=>Y	0	10	0.6
89 IF X<=>F	0	12	0.4
90 LOOP INDEX	0	3	0.5
91 IF FLAG/PORT	0	6	0.3
92 IF TIME	1	11	0.3
93 BEGIN CASE	1	8	0.2
94 ELSE	0	4	0.2
95 END	0	4	0.2
96 SERIAL OUT	0	3	Option: 0x 1x 2x 3x Time: 0.4 1.8 2.1 0.9 Option: 4x 5x 6x 7x Time: 1.7 1.9 0.7 0.5
97 INIT.TELE.	7	17	2.3
98 SEND CHAR.	0	3	0.7+0.05 * No. char.

3.10 ERROR CODES

There are four types of errors flagged by the CR10: Compile, Run Time, Editor, and *D Mode. Compile errors are errors in programming which are detected once the program is entered and compiled for the first time (*0, *6, or *B Mode entered). If a programming error is detected during compilation, an E is displayed with the 2 digit error code. The Instruction Location Number of the Instruction which caused the error is displayed to the right of the error code (e.g., 105 indicates that the fifth instruction in Table 1 caused the error). Error 22, missing END, will indicate the location of the instruction which the compiler cannot match with the END instruction.

Run time errors are detected while the program is running. The number of the instruction being executed at the time the error is detected is displayed to the right of the error code (e.g., E09 06 indicates that an Instruction 6 in the program is attempting to store data in input locations beyond those allocated). Run time errors 9 and 31 are the result of programming errors. While error 8 will display the number of the instruction that was being executed when the error occurred, it is unlikely that the instruction has anything to do with the error.

If there is a run time error in a table with a fast execution interval, the error may be written to the display so frequently that it seems the CR10 is not responding to the keyboard. Once the program is stopped, normal function will return. To stop the program some entry must be changed which requires recompiling (Section 1.1.4). For example, enter 0 for the execution interval of Table 1 (i.e., enter *1A0A as fast as possible).

Error 8 is the result of a hardware and software "watchdog" that checks the processor state, software timers, and program related counters. The watchdog will attempt to reset the processor and program execution if it finds that the processor has bombed or is neglecting standard system updates, or if the counters are out of allowable limits. Error code 08 is flagged when the watchdog performs this reset. E08 is occasionally caused by voltage surges or transients. Frequent repetitions of E08 are indicative of a hardware problem or a software bug and should be reported to Campbell Scientific. The CR10 keeps track of the number of times (up to 99) that E08 has occurred. The number can be displayed and reset in the *B Mode (Section 1.6) or with the Telecommunications A command (Section 5.1).

Editor errors are detected as soon as an incorrect value is entered and are displayed immediately. Only the error code is displayed.

*D Mode errors indicate problems with saving or loading a program. Only the error code is displayed.

TABLE 3.10-1. Error Codes

Code	Type	Description
03	Editor	program table full
04	Compile	Intermediate Storage full
05	Compile	Storage Area #2 not allocated
08	Run Time	CR10 reset by watchdog timer
09	Run Time	Insufficient Input Storage
11	Editor	Attempt to allocate more Input or Intermediate Storage than is available
20	Compile	SUBROUTINE encountered before END of previous subroutine
21	Compile	END without IF, LOOP or SUBROUTINE
22	Compile	Missing END
23	Compile	Nonexistent SUBROUTINE
24	Compile	ELSE in SUBROUTINE without IF
25	Compile	ELSE without IF
26	Compile	EXIT LOOP without LOOP
27	Compile	IF CASE without BEGIN CASE
30	Compile	IF and/or LOOP nested too deep
31	Run Time	SUBROUTINES nested too deep
40	Editor	Instruction does not exist
41	Editor	Incorrect execution interval
60	Compile	Inadequate Input Storage for FFT
61	Compile	Burst Measurement Scan Rate too short
96	*D MODE	Addressed device not connected
97	*D MODE	Data not received within 30 seconds
98	*D MODE	Uncorrectable errors detected
99	*D MODE	Wrong file type or editor error

SECTION 4. EXTERNAL STORAGE PERIPHERALS

External data storage devices are used to provide a data transfer medium that the user can carry from the test site to the lab and to supplement the internal storage capacity of the CR10, allowing longer periods between visits to the site. The standard data storage peripheral for the CR10 is the Storage Module (Section 4.5). Output to a printer or related device is also possible (Section 4.4).

*Data output to a peripheral device can take place ON-LINE (automatically, as part of the CR10's routine operation) or it can be MANUALLY INITIATED. On-line data transfer is accomplished with Instruction 96 (Section 4.1). Manual initiation is done in the *8 Mode (Section 4.2).*

The CR10 can output data to multiple peripherals. The CR10 activates the peripheral it sends data to in one of two ways (Section 6.2):

1. *A specific pin in the 9-pin connector is dedicated to that peripheral; when that pin goes high, the peripheral is enabled. This is referred to as "PIN-ENABLED" or simply "ENABLED".*

2. *The peripheral is synchronously addressed by the CR10. This is referred to as "ADDRESSED".*

Cassette tape and modems are pin-enabled. Only one cassette recorder and only one modem/terminal device may be connected to the CR10 at any one time.

The SM192 and SM716 Storage Modules are addressed. The CR10 can tell when the addressed device is present. The CR10 will not send data meant for the Storage module if the Storage Module is not present (Section 4.5.2).

*The *9 Mode (Section 4.6) allows the user to communicate directly with the Storage Module and to perform several functions, including review of data, battery test, review of Storage Module status, etc.*

4.1 ON-LINE DATA TRANSFER - INSTRUCTION 96

All on-line data output to a peripheral device is accomplished with Instruction 96. (Instruction 96 can also be used to transfer data from one Final Storage Area to the other, Section 12). This instruction must be included in the datalogger program for on-line data transfer to take place. Instruction 96 should follow the Output Processing Instructions, but only needs to be included once in the program tables unless both Final Storage areas are in use. The suggested programming sequence is:

1. Set the Output Flag.

2. If both Final Storage Areas are in use or if you wish to set the Output Array ID, enter Instruction 80 (Section 10).

3. Enter the appropriate Output Processing Instructions.

4. Enter Instruction 96 to enable the on-line transfer of Final Storage data to the specified device. If outputting to more than one device, Instruction 96 must be entered separately for each device.

5. Repeat steps 2 through 4 if you wish to output data to the other Final Storage Area and the peripheral.

Instruction 96 has a single parameter which specifies the peripheral to send output to. Table 4.1-1 lists the output device codes.

TABLE 4.1-1. Output Device Codes for Instruction 96 and *8 Mode

Code	Device
00	Tape. Data transferred in blocks of 512 Final Storage locations
09	Tape. All data since last output. [Inst. 96 only]
	ADDRESSED PRINTER
1x	Printable ASCII
2x	Comma delineated ASCII
3x	Binary
	PIN ENABLED PRINTER
4x	Printable ASCII
5x	Comma delineated ASCII
6x	Binary
	x = BAUD RATE CODES 0 300 1 1200 2 9600 3 76,800
7N	Storage Module N (N=address, 1...8)
7N--	Output File Mark to Storage Module N
80	To the other Final Storage Area [Inst. 96 only], new data since last output
81	To the other Final Storage Area [Inst. 96 only], entire active Final Storage Area

The source of data for Instruction 96 is the currently active Final Storage Area as set by Instruction 80 (the default is Final Storage Area 1 at the beginning of each program table execution).

If the CR10 is using the 9 pin connector for other I/O tasks when Instruction 96 is executed, the output request is put in a queue and program execution continues. As the 9-pin connector becomes available, each device in the queue gets its turn.

An output request is not put in the queue if the same device is already in the queue. The data contained in the queue (and which determine a unique entry) are the device, baud rate (if applicable), and the Final Storage Area.

When an entry reaches the top of the queue, the CR10 sends all data accumulated since the last transfer to the device up to the location of the DSP at the time the device became active.

The most efficient use of cassette tape and power is made with the CASSETTE TAPE option to transfer data in blocks of 512 Final Storage locations. (Data is always written in the equivalent of 512 locations. If code 09 was used, and there are only 10 new values, sending this data would include 502 null characters.)

Option 09, transfer any new data, is used if it is desired to run the tape only at particular times or under certain conditions (the program is written so that 96 only gets executed when these conditions are met). When 96 finally does get executed, all data between the TPTR and DSP, including a final block less than 512 locations, are written to tape.

Section 4.3 contains specifics on the cassette recorder. Note that tape operation is for above freezing temperatures only.

Printer output can be either pin-enabled or addressed. However, there is not a pin specifically dedicated to print enable. When a pin-enabled print output is specified, the SDE line, which is normally used in the addressing sequence, is used as a print enable. This allows some compatibility with the CR21, 21X, and CR7 dataloggers which have a Print Enable line. The pin-enabled print option will result in garbage being sent to the print peripheral if an addressed device is also connected to the CR10 (i.e., CR10KD, SM192 or SM716 etc.). The SDC99 Synchronous Device Interface can convert a print device to an Addressed peripheral (Section 6.2).

The STORAGE MODULE address is important only when using more than one Storage Module, 1 is a universal address which will find the Storage Module with lowest number address that is connected. If a Storage Module is not connected, the CR10 will not advance the SPTR (Section 2.1) and the Storage Module drops out of the queue until the next time Instruction 96 is executed. Section 4.5 contains specifics on the SM192 and SM716.

TABLE 4.2-1. *8 Mode Entries

Key	Display ID:DATA	Description
*8	08:00	Key 1 or 2 for Storage Area. (This window is skipped if no memory has been allocated to Final Storage Area 2.)
A	01:XX	Key in Output Device Option. See Table 4.1-1.
A	02:XXXXX	Start of dump location. Initially the TPTR, SPTR or PPTR location; a different location may be entered if desired.
A	03:XXXXX	End of dump location. Initially the DSP location; a different location may be keyed in if desired.
A	04:00	Ready to dump. To initiate dump, key any number, then A. While dumping, "04" will be displayed in the ID field and the location number in the Data field. The location number will stop incrementing when the dump is complete. (Any key aborts transmission after completion of the current data block.)

4.2 MANUALLY INITIATED DATA OUTPUT - *8 MODE

Data transfer to a peripheral device can be manually initiated in the *8 Mode. This process requires that the user have access to the CR10 through a terminal or the Campbell Scientific Keyboard/Display. The *8 Mode allows the user to retrieve a specific block of data, on demand, regardless of whether or not the CR10 is programmed for on-line data output.

If external storage peripherals (cassette, Storage Module, etc.) are not left on-line, the maximum time between site visitations and data retrieval must be calculated to ensure that data placed in Final Storage are not lost due to write-over. In order to make this calculation, users must determine: (1) how large their Final Storage is, (2) how many Output Arrays are being generated, (3) how many low and/or high resolution data points are included per Output Array, and (4) the rate at which Output Arrays are placed into Final Storage. When calculating the number of data points per Output Array, remember to add 1 overhead data point (2 bytes) per array for the Output Array ID.

For example, assume that 29900 locations are assigned to Final Storage (*A Mode), and that 1 Output Array, containing the Array ID (1 memory location), 9 low resolution data points (9 memory locations) and 5 high resolution data points (10 memory locations), is stored each hour. In addition, an Output Array with the Array ID and 5 high resolution data points (11 memory locations) is stored daily. This is a total of 491 memory locations per day ((20 x 24) + 11). 29900 divided by 491 = 60.90 days. Therefore, the CR10 would have to be visited every 60 days to retrieve data, because write-over would begin on the 61st day. Most likely the user would want to retrieve data more often than this to perform a general checkout of the station.

The output device codes used with the *8 Mode are the same as those used with Instruction 96 (Table 4.1-1), with the exception of "all data to tape" (09, with *8 all data between the start and stop locations is always written) and the options to transfer data from one Final Storage area to the other (80, 81). Table 4.2-1 lists the keystrokes required to initiate a *8 data dump.

4.3 CASSETTE TAPE OPTION

The Model RC35 Cassette Tape Recorder or equivalent can be left attached to the CR10 for continuous on-line data recording or it can be periodically taken to the CR10 site for the manually initiated retrieval of the data accumulated in Final Storage. The *8 Mode is used to manually initiate tape transfer.

4.3.1 CASSETTE RECORDER

The RC35 Cassette Recorder offered by Campbell Scientific is an inexpensive recorder for use with the CR10 (also compatible with the 21X and CR7 dataloggers). The record/playback function of each RC35 is tested along with a head alignment procedure prior to shipment. CR10/RC35 connections are made with the SC92A Cassette Write Only Interface or the SC93A Cassette Read/Write Interface. The CR10 controls the on/off state of

the RC35 by switching power through the DC power line of the SC92A/SC93A.

TABLE 4.3-1 Cassette Recorder Specifications

Power	6 VDC (provided by CR10 through SC92A or SC93A); 4 AA size batteries; 120 VAC/6 VDC adapter
Current Drain while Recording	200 mA typ./5 sec., 300 max.
Tape Length	C-60 recommended
Tape Quality	Normal bias, high quality (e.g., TDK, Maxell)
External Inputs	Mic., DC In, Monitor, and Remote
Operating Temperature	0° to +40°C

POWER SUPPLY

The CR10's internal power supply will power the recorder during periods of data transfer, but will NOT be available to play, advance, or back-up tapes. In order to perform these functions during setup and check-out operations, the recorder requires 4 alkaline AA batteries or the 120 VAC adapter.

OPERATING TEMPERATURE LIMITATIONS

The cassette recorder is recommended for use in an environmental operating temperature range of 0° to +40°C. Temperatures below 0°C may cause tape speed variation in excess of that which can be tolerated during playback. If the RC35 is outside the 0°C to 40°C range, data transferred may be unreadable.

VOLUME CONTROL

When recording data, the RC35's volume setting does not matter. The recorder is equipped with an automatic gain control which controls the recorded signal level. For playback, a mid- range volume setting is normally required.

CASSETTE TAPES

Normal bias, high quality cassette tapes are recommended for use with the recorder. The more expensive high bias chromium oxide tapes will NOT perform satisfactorily. Although the use of C-90 tapes is generally successful, Campbell Scientific recommends the use of C-60 (30 minutes per side) cassettes. TDK, Maxell, and equivalent quality cassette tapes perform well and are readily available. Bargain-priced tapes have often performed poorly and are not recommended.

New tapes are often tightly wound, creating enough drag or pressure to cause the tape recorder to "pop" out of the record mode. This potential loss of data may be overcome by fast-forward/rewinding the entire tape before placing it in service.

4.3.2 CASSETTE CONNECTOR INTERFACE CABLES

A cassette interface cable is required to connect the cassette recorder to the CR10. Two models are available. The SC92A is a WRITE ONLY interface. The SC93A is a READ/WRITE interface that allows the CR10 to load datalogger programs from tape in addition to writing data and programs. The SC93A is required only if special software exists in the datalogger PROM for transferring programs via tape (refer to Appendix B)

The SC92A and SC93A have a combination backshell circuit card and subminiature 9-pin D-type connector which attaches to the socket connector on the wiring panel. The other end of the SC92A has two plugs which are plugged into the POWER and MIC jacks on the recorder. The SC93A has three plugs which are plugged into the POWER, MIC and EAR (or MONITOR) jacks on the recorder. Both cables transform 12 V from the CR10 to 6 V for powering the recorder during periods of data transfer. Additional circuitry shapes the data signal waveform.

WARNING: The SC92/SC93 interfaces previously supplied with the 21X and CR7 dataloggers are not compatible with the CR10. The SC235 CR21 Cassette Connector Interface supplied with the CR21 datalogger is not compatible with the CR10. If the SC92, SC93, or SC235 interfaces are used with the CR10, the data on tape CANNOT be recovered!

4.3.3 TAPE FORMAT

Data is transferred to cassette tape in the high speed/high density Format 2. Data tapes generated by the CR10 are read by the PC201 tape read card for the IBM PC or by the C20 Cassette Interface. The C20 decodes the tape and transmits the data in ASCII to any external device equipped with a standard RS232 interface.

TABLE 4.3-2. Format 2 Specifications

Data	Binary
Low Resolution	2 bytes/data point
High Resolution	4 bytes/data point
C-60 Capacity (Lo Res.)	180,000 data points (1 side only)
Data Transfer Rate (Lo Res.)	100 data points/sec.
Block Size	512 Final Storage locations

4.3.4 CONNECTING TAPE TO CR10

The procedure for setting up the CR10 and cassette recorder for transfer to tape is as follows:

1. Load a cassette in the recorder and advance the tape forward until the tape leader is past the recording head. (Internal batteries or AC power required.)
2. Connect the SC92A or SC93A to the 9-pin D-TYPE connector in the upper right-hand corner of the wiring panel. (Via the SC12 ribbon cable if using *8 with CR10KD or modem/terminal.)
3. Connect the plugs on the free end of the SC92A or SC93A into the DC-IN and MIC (and Ear if SC93A) jacks on the recorder.
4. Simultaneously press the RECORD and PLAY buttons on the recorder to set it for recording. With the DC-IN Jack plugged in, the tape will not move until the dump occurs.
5. To test connections manually initiate transfer by Keying in the *8 commands as listed in Table 4.2-1. The tape should advance as data is transferred. If the Start of dump location is equal to the End of dump location, the CR10 will write a "dummy" block of data to tape.

If you are leaving the recorder with the CR10 (on-line output to tape enabled with Instruction 96) it is a good idea to write a dummy block of data to tape (5 above) to ensure that the recorder is correctly connected. Leave the CR10 in the *0 Mode.

When on-line, the CR10 dumps data to tape in 512 location blocks (unless the option to dump any new data is selected in Instruction 96). When picking up a data tape from a field site, dump the residual data (data which has accumulated since the last full block) before removing the tape. Dump the residual data by entering the *8 Mode, advancing through windows 2 and 3 and initiating a dump. (The start and stop locations should be less than 512 locations apart.) After removing the old tape, insert a new tape and go through the set up steps above.

4.4 PRINTER OUTPUT FORMATS

Printer output can be sent in Final Storage Format (Appendix C.2), Printable ASCII, or Comma Delineated ASCII. These ASCII formats may also be used when data from the Storage Modules or Telecommunications are stored on disk with Campbell Scientific's PC208 software.

4.4.1 PRINTABLE ASCII FORMAT

In the Printable ASCII format each data point is preceded by a 2 digit data point ID and a (+) or (–) sign. The ID and fixed spacing of the data points make particular points easy to find on a printed output. This format requires 10 bytes per data point to store on disk.

Figure 4.4-1 shows both high and low resolution data points in a 12 data point Output Array. The example data contains Day, Hour-Minute, and Seconds in the 2nd - 4th data points. REMEMBER! You must specifically program the CR10 to output the date and time values. The Output Array ID, Day, and Time are always 4 character numbers, even when high resolution output is specified. The Seconds resolution is .125 seconds.

Each full line of data contains 8 data points (79 characters including spaces), plus a carriage return (CR) and line feed (LF). If the last data point in a full line is high resolution, it is followed immediately with a CR and LF. If it is low resolution, the line is terminated with a space, CR and LF. Lines of data containing less than 8 data points are terminated similarly after the last data point.

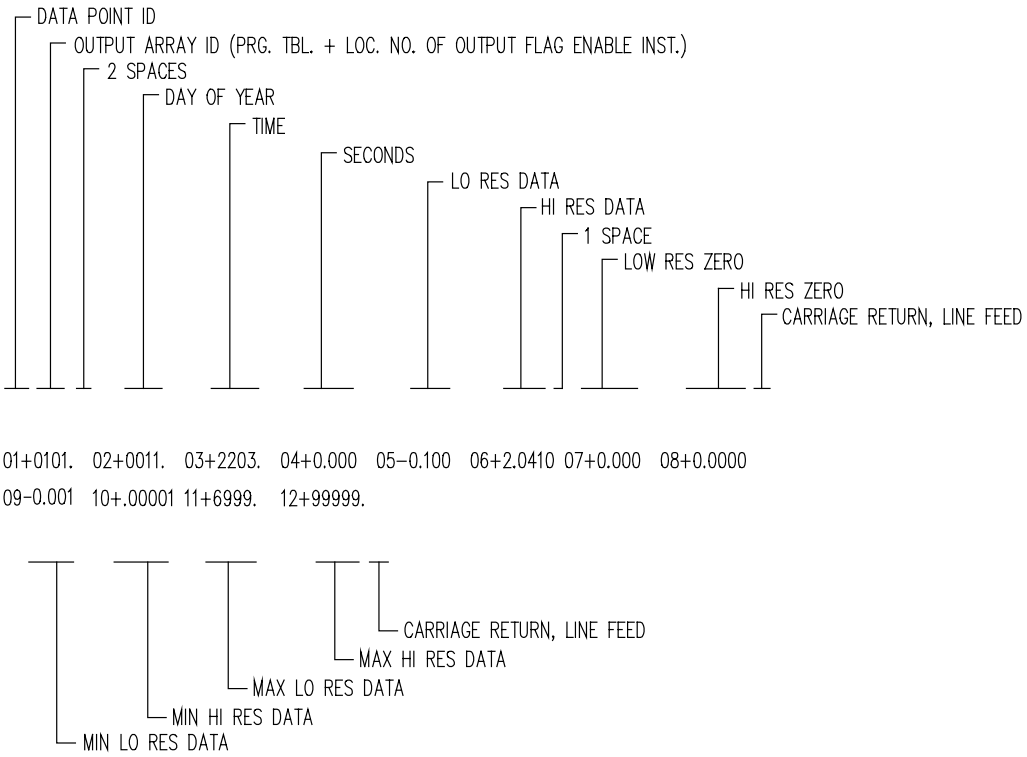


FIGURE 4.4-1. Example of CR10 Printable ASCII Output Format

4.4.2 COMMA DELINEATED ASCII

Comma Delineated ASCII strips all IDs, leading zeros, unnecessary decimal points and trailing zeros, and plus signs. Data points are separated by commas. Arrays are separated by Carriage Return Line Feed. Comma Delineated ASCII requires approximately 6 bytes per data point. Example:

1,234,1145,23.65,-12.26,625.9
1,234,1200,24.1,-10.98,650.3

4.5 STORAGE MODULE (SM192/716)

The Storage Module stores data in battery backed RAM. Backup is provided by an internal lithium battery. Operating power is supplied by the CR10 over pin 1 of the 9-pin connector. When power is applied to the Storage Module, a File Mark is placed in the data (if a File Mark is not the last data point already in storage).

The File Mark separates data. For example, if you retrieve data from one CR10, disconnect the Storage Module and connect it to a second CR10, a File Mark is automatically placed in the data. This mark follows the data from the first CR10 but precedes the data from the second.

The SM192 has 192K bytes of RAM storage; the SM716 has 716K bytes. Both can be configured as either ring or fill and stop memory.

4.5.1 STORAGE MODULE ADDRESSING

The capability of assigning different addresses to Storage Modules allows 1) multiple (up to 8) Storage Modules to be connected to the CR10 during on-line output (Instruction 96), 2) different data to be output to different Modules, and 3) transfer of data from a Module that is left with the CR10 to a Module that is hand carried to the site for data transfer (*9 Mode).

Storage Modules are assigned addresses (1-8) either through the *9 Mode or through Telecommunications (SM192/SM716 Manual). 1 is the default address when the Storage Module is reset, and unless you are using one of the features which require different addresses you need not assign any other address.

Address 1 is also a universal address when sending data or commands to a storage module with Instruction 96, *8, or *9. When address 1 is entered in the *9 Mode (default) or in the device code (71, Table 4.2-1) for Instruction 96 or the *8 Mode, The CR10 searches for the Storage Module with the lowest address that is not full (fill and stop configuration only) and addresses it. In other words, if a single Storage

Module is connected, and it is not full, address 1 will address that Storage Module regardless of the address that is assigned to the Module.

Address 1 would be used with Instruction 96 if several Storage Modules with different addresses were connected to the CR10 and were to be filled sequentially. The Storage modules would be configured as fill and stop. When the lowest addressed Module was full data would be written to the next lowest addressed Module, etc.

4.5.2 STORAGE MODULE USE WITH INSTRUCTION 96

When output to the Storage Module is enabled with Instruction 96, the Storage Module(s) (see 4.5.1 for addressing on multiple modules) may be either left with the CR10 for on-line data transfer and periodically exchanged, or brought to the site for data transfer.

USE OF STORAGE MODULE TO PICK UP DATA

The CR10 is capable of recognizing whether or not the Storage Module is connected. Each time Instruction 96 is executed and there is data to output, the CR10 checks for the presence of the Storage Modules. If one is not present, the CR10 does not attempt to output data to it. Instead, the CR10 saves the data and continues its other operations without advancing the Storage Module Pointer (SPTR, Section 2.1).

When the user finally does connect the Storage Module to the CR10, two things happen:

1. Immediately upon connection, a File Mark is placed in the Storage Module Memory following the last data stored (if a File Mark wasn't the last data point already in storage).
2. During the next execution of Instruction 96, the CR10 recognizes that the Storage Module (SM) is present and outputs all data between the SPTR and the DSP location.

The File Mark allows the operator to distinguish blocks of data from different dataloggers or from different visits to the field.

To be certain that the SM has been connected to the CR10 during an execution of P96, the user can:

- Leave the SM connected for a time period longer than an execution interval OR
- Use the SC90 9-Pin Serial Line Monitor. The SC90 contains an LED which lights up during data transmission. The user connects the SM to the CR10 with the SC90 on the line and waits for the LED to light. When the light goes off, data transfer is complete and the SM can be disconnected from the CR10.

4.5.3 *8 DUMP TO STORAGE MODULE

In addition to the on-line data output procedures described above, output from CR10 Final Storage to the SM192 and SM716 can be manually initiated in the *8 Mode. The procedure for setting up and transferring data is as follows:

1. Connect the CR10KD Keyboard/Display (or terminal) and the Storage Module in parallel to the CR10 using the SC12 cable. For terminals, an SC32A will be needed. See Section 5 for interfacing details.
2. Key in the appropriate commands as listed in Table 4.2-1.

4.6 *9 MODE -- STORAGE MODULE COMMANDS

The *9 Mode is used to issue commands to the Storage Module (from the CR10) using the CR10KD or a terminal/computer. These "commands" are like * Modes for the Storage Module and in some cases are directly analogous to the CR10 * Modes. Command 7 enters a mode used to review stored data, and 8 is used to transfer data between two Storage Modules connected to the CR10. The operations with the Storage Module are not directly analogous as may be seen in Table 4.6-1 which lists the commands (e.g., when reviewing data, #A advances to the start of the next Output Array rather than to the same element in the next array with the same ID).

When *9 is keyed, the CR10 responds: 09:01

1 is the default address for the Storage Module (Section 4.4.1). If you have more than 1 Storage Module connected, enter the address of the desired Storage Module. Address 1 will always work if only one Module is connected. Key A and the CR10 responds: 9N:00

Where N is the address which was entered. You may now enter any of the commands in Table 4.6-1 (key in the command number and enter with A). Most commands have at least

one response, advance through these and return to the *9 command state by keying A.

TABLE 4.6-1. *9 Commands for Storage Module

COMMAND	DISPLAY	DESCRIPTION
1	01: 0000	RESET, enter 248 to erase all data and programs. While erasing, the SM checks memory. The number of good chips is then
	01: XX	displayed (6 for SM192, 22 SM716).
3	03: 01	INSERT FILE MARK, 1 indicates that the mark was inserted, 0 that it was not.
4	04: XX	DISPLAY/SET MEMORY CONFIGURATION enter the appropriate code to change configuration 0=ring, 1=fill & stop
5		DISPLAY STATUS (A to advance to each window)
	01: ABCD	Window 1:
	AB	Storage pointer location (chip no.)
	CD	Total good RAM chips (1-22)
	02: ABCD	Window 2:
	AB	Display pointer location (chip no.)
	C	Unloaded Batt. Chk. 0=low, 1=OK
	D	No. of Programs stored (Max=9)
	03: A0CD	Window 3:
	A	Errors logged (up to 9)
	0	Not Used
	C	Memory Config. (0=ring, 1=fill&stop)
	D	Memory Status (0=not full, 1=full)
	04: XXXXX	PROM signature (0 if bad PROM)
6	06: 0X	BATTERY CHECK UNDER LOAD (0=low, 1=OK)
7	07: 00	DISPLAY DATA, Select the Storage Module Area with the these codes: 0 Dump pointer to SRP 1 File 1, current file 2 File 2, previous to file 1 3 File 3, previous to file 2 4 File 4, previous to file 3 5 File 5, previous to file 4 7 Display pointer to SRP 9 Oldest data to SRP 1-5 will loop within file boundaries, 0,7,9 allow display to cross boundaries
	07:XXXXXX	SM location at end of area selected. Key A to advance to first data. If another location is keyed in SM will jump to 1st start of array following that location. Review data with: A Advance and display next data point B Back-up one data point # Display location, C to return to data #A Advance to next start of Array #B Back-up to start of Array #D Return to *9 command mode
8		DUMP TO ANOTHER STORAGE MODULE
	08:00	Select Area as in 7 above
	01:XXXXXX	First Loc. in area selected/Enter Loc. to start dump
	02:XXXXXX	Final Loc. in area selected/Enter Loc. to end dump
	03:XX	Enter destination SM address
9		DISPLAY ADDRESSES OF CONNECTED SM
	XXXXXXXX	1 = occupied, 0 = unoccupied
	87654321	(Addresses 8-1 from left to right)
10		CHANGE ADDRESS

10:0X	X is current address, enter address to change to (1-8)

SECTION 5. TELECOMMUNICATIONS

Telecommunications is used to retrieve data from Final Storage directly to a computer/terminal and to program the CR10. Any user communication with the CR10 that makes use of a computer or terminal instead of the CR10KD is through Telecommunications.

Telecommunications can take place over a variety of links including:

- *Telephone*
- *Cellular phone*
- *Radio frequency*
- *Short haul modem and twisted pair wire*
- *SC32A and ribbon cable*
- *Multi-drop interface and coax cable*

This section does not cover the technical interface details for any of these links. Those details are covered in Section 6 and in the individual manuals for the devices.

Data retrieval can take place in either ASCII or BINARY. The BINARY format is 5 times more compact than ASCII. The shorter transmission times for binary result in lower long distance costs if the link is telephone and lower power consumption with an RF link. On "noisy" links shorter blocks of data are more likely to get through without interruption.

For more efficient data transfer, binary data retrieval makes use of a signature for error detection. The signature algorithm assures a 99.998% probability that if either the data or its sequence changes, the signature changes.

Campbell Scientific has developed a software package which automates data retrieval and facilitates the programming of Campbell Scientific dataloggers and the handling of data files. This package (PC208) has been designed to meet the most common needs in datalogger support and telecommunications. Therefore, this section does not furnish sufficient detail to write telecommunications software. Appendix C contains some details of binary data transfer and Campbell Scientific's binary data format.

The emphasis of this section is on the commands that a person would use when manually (i.e., keyed in by hand) interrogating or programming the CR10 via a computer/terminal. These commands and the responses to them are sent in the American Standard Code for Information Interchange (ASCII).

The telecommunications commands allow the user to perform several operations including:

- *monitor data in Input Storage- review data in Final Storage*
- *retrieve Final Storage data in either ASCII or BINARY*
- *open communications with the Storage Module*
- *remote keyboard programming*

The Remote Keyboard State (Section 5.2) allows the user with a computer/terminal to use the same commands as the CR10KD.

5.1 TELECOMMUNICATIONS COMMANDS

When a modem/terminal rings the CR10, the CR10 should answer almost immediately. Several carriage returns (CR) must be sent to the CR10 to allow it to set its baud rate to that of the modem/terminal (300, 1200, 9600, or 76,800). Once the baud rate is set, the CR10 will send back the prompt, "*****", signaling that it is ready to receive a command.

GENERAL RULES governing the telecommunications commands are as follows:

1. ***** from datalogger means "ready for command".
2. All commands are of the form: [no.]letter, where the number may or may not be optional.

3. Valid characters are the numbers **0-9**, the capital letters **A-M**, the colon (**:**), and the carriage return (**CR**).
4. An illegal character increments a counter and zeros the command buffer, returning a *****.
5. **CR** to datalogger means "execute".
6. **CRLF** from datalogger means "executing command".
7. ANY character besides a CR sent to the datalogger with a legal command in its buffer causes the datalogger to abort the command sequence with **CRLF*** and to zero the command buffer.
8. All commands return a response code, usually at least a checksum.
9. The checksum includes all characters sent by the datalogger since the last *****, including the echoed command sequence, excluding only the checksum itself. The checksum is formed by summing the ASCII values, without parity, of the transmitted characters. The largest possible checksum value is 8191. Each time 8191 is exceeded, the CR10 starts the count over; e.g., if the sum of the ASCII values is 8192, the checksum is 0.
10. Commands that return Campbell Scientific binary format data (i.e., **F** and **K** commands) return a signature (see Appendix C.3).

The CR10 sends ASCII data with 8 bits, no parity, one start bit, and one stop bit.

After the CR10 answers a ring, or completes a command, it waits about 40 seconds (127 seconds in the Remote Keyboard State) for a valid character to arrive. It "hangs up" if it does not receive a valid character in this time interval. Some modems are quite noisy when not on line; it is possible for valid characters to appear in the noise pattern. To insure that this situation does not keep the CR10 in telecommunications, the CR10 counts all the invalid characters it receives from the time it answers a ring, and terminates communication after receiving 150 invalid characters.

The CR10 continues to execute its measurement and processing tasks while servicing the telecommunication requests. If the processing overhead is large (short Execution Interval), the processing tasks will slow the telecommunication functions. In a worst case situation, the CR10 interrupts the processing tasks to transmit a data point every 0.125 second.

The best way to become familiar with the Telecommunication Commands is to try them from a terminal connected to the CR10 via the SC32A (Section 6.7.1) or other interface. Commands used to interrogate the CR10 in the Telecommunications Mode are described in the following Table.

TABLE 5.1-1. Telecommunications Commands

Command	Description
[F.S. Area]**A**	SELECT AREA/STATUS - If 1 or 2 does not precede the A to select the Final Storage Area, the CR10 will default to Area 1. All subsequent commands other than A will address the area selected. Datalogger returns Reference, the DSP location; the number of filled Final Storage locations; Version of datalogger; Errors #1 and #2 where #1 is the number of E08's and #2 is the number of overrun errors (both are cleared by entering 8888A; #2 is also cleared at time of program compilation); size of total Memory in CR10; Final Storage Area; Location of MPTR; and Checksum. All in the following format: R+xxxxx Fxxxxx Vxx Exx xx Mxx Ax L+xxxxx Cxxxx If data is stored while in telecommunications, the A command must be issued to update the Reference to the new DSP.
[no. of arrays]**B**	BACK-UP - MPTR is backed-up the specified number of Output Arrays (no number defaults to 1) and advanced to the nearest start of array. CR10 sends the Area, MPTR Location, and Checksum: Ax L+xxxxx Cxxxx
[YR:DAY:HR:MM:SS]**C**	RESET/SEND TIME - If time is entered the time is reset. If only 2 colons are in the time string, HR:MM:SS is assumed; 3 colons means DAY:HR:MM:SS. If only the C is entered, time is unaltered. CR10 returns year, Julian day, hr:min:sec, and Checksum: Y:xx Dxxxx Txx:xx:xx Cxxxx
[no. of arrays]**D**	ASCII DUMP - If necessary, the MPTR is advanced to start of scan. CR10 sends the number of arrays specified (no number defaults to 1) or the number of arrays between MPTR and Reference, whichever is smaller, CRLF, Location, Checksum.
E	End call. Datalogger sends CRLF only.
[no. of loc.]**F**	BINARY DUMP - Used in TELCOM (PC208). See Appendix C.
[F.S. loc. no.]**G**	MOVE MPTR - MPTR is moved to specified Final Storage location. The location number must be entered. CR10 sends Area, Location, and Checksum: Ax L+xxxxx Cxxxx
7H or **2718H**	REMOTE KEYBOARD - CR10 sends the prompt ">" and is ready to execute standard keyboard commands (Section OV3).
[loc. no.]**I**	Display/change value at Input Storage location. CR10 sends the value stored at the location. A new value and CR may then be sent. CR10 sends checksum. If no new value is sent (CR only) the location value will remain the same.
3142J	TOGGLE FLAGS AND SET UP FOR K COMMAND - Used in the Monitor Mode and with the Heads Up Display. See Appendix C for details.

K	CURRENT INFORMATION - In response to the K command, the CR10 sends datalogger time, user flag status, the data at the input locations requested in the J command, and Final Storage Data if requested by the J command. Used in the Monitor Mode and with Heads Up Display. See Appendix C.
[Password]**L**	Unlocks security (if enabled) to the level determined by the password entered (See *C Mode, Section 1.7). CR10 sends security level (0-3) and checksum: Sxx Cxxxx
[X]**M**	Connect to Storage Module with address 'X' and enter the Storage Module's Telecommunications Mode (see Storage Module manual). The Storage Module can also be accessed through the *9 Commands while in the Remote Keyboard (Section 4.6 and the Storage Module manual).
1N	Connect phone modem to RF modem at phone to RF base station.

5.2 REMOTE PROGRAMMING OF THE CR10

Remote programming of the CR10 can be accomplished with the PC208 software or directly through the Remote Keyboard State.

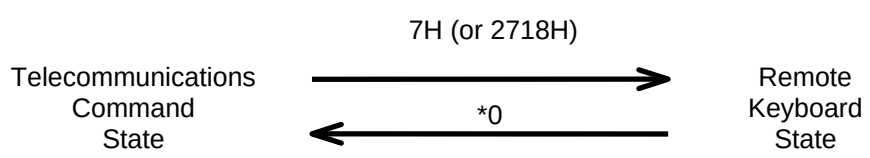
The PC208 Datalogger Support Software was developed by Campbell Scientific for use with IBM or compatible PC's. Datalogger programs are developed on the computer using the program editor (EDLOG) and downloaded to the datalogger with the terminal emulator program (GraphTerm).

The CR10 is placed in the Remote Keyboard State by sending either "**7H**" or "**2718H**" and a carriage return (CR). The CR10 responds by sending a CR, line feed (LF), and the prompt '>'. The CR10 is then ready to receive the standard keyboard commands; it recognizes all the standard CR10 keyboard characters plus several additional characters, such as the decimal point and the minus sign (Section OV3.2). ENTERING *0 RETURNS THE CR10 TO THE TELECOMMUNICATIONS COMMAND STATE.

Remember that entering *0 will compile and run the CR10 program if program changes have been made. If the CR10KD is connected it will just display "LOG" when *0 is executed via telecommunications. It will not indicate active tables (keying "*0" on the Keyboard/Display will show the tables).

The 7H Command is generally used with a terminal for direct entry since H makes use of a destructive backspace and does not send control Q between each entry. The 2718H Command functions the same as it does for other Campbell Scientific dataloggers (deleting an entry causes the entire entry to be sent, "control Q" is sent after each user entry) and its use will be familiar to those already working with a 21X or CR7X.

It is important to remember that the Remote Keyboard State is still within Telecommunications. Entering *0 exits the Remote Keyboard and returns the datalogger to the Telecommunications Command State, awaiting another command. So, the user can step back and forth between the Telecommunications Command State and the Remote Keyboard Mode.



SECTION 6. 9-PIN SERIAL INPUT/OUTPUT

6.1 PIN DESCRIPTION

All external communication peripherals connect to the CR10 through the 9-pin subminiature D-type socket connector located on the front of the Wiring Panel (Figure 6.1-1). Table 6.1-1 shows the I/O pin configuration, and gives a brief description of the function of each pin.

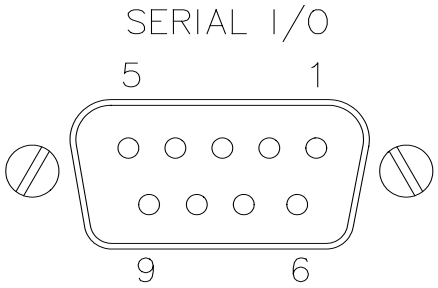


FIGURE 6.1-1. 9-pin Female Connector

TABLE 6.1-1. Pin Description

ABR = Abbreviation for the function name.
PIN = Pin number.
O = Signal Out of the CR10 to a peripheral.
I = Signal Into the CR10 from a peripheral.

PIN	ABR	I/O	Description
1	5 V	O	5V: Sources 5 VDC, used to power peripherals.
2	SG		Signal Ground: Provides a power return for pin 1 (5V), and is used as a reference for voltage levels.
3	RING	I	Ring: Raised by a peripheral to put the CR10 in the telecommunications mode.
4	RXD	I	Receive Data: Serial data transmitted by a peripheral are received on pin 4.
5	ME	O	Modem Enable: Raised when the CR10 determines that a modem raised the ring line.
6	SDE	O	Synchronous Device Enable: Used to address Synchronous Devices (SDs), and can be used as an enable line for printers.
7	CLK/HS	I/O	Clock/Handshake: Used with the SDE and TXD lines to address and transfer data to SDs. When not used as a clock, pin 7 can be used as a handshake line (during printer output, high enables, low disables).
8	TE	O	Tape Enable: Powers the cassette recorder during tape transfer.
9	TXD	O	Transmit Data: Serial data are transmitted from the CR10 to peripherals on pin 9; logic low marking (0V) logic high spacing (5V) standard asynchronous ASCII, 8 data bits, no parity, 1 start bit, 1 stop bit, 300, 1200, 9600, 76,800 baud (user selectable).

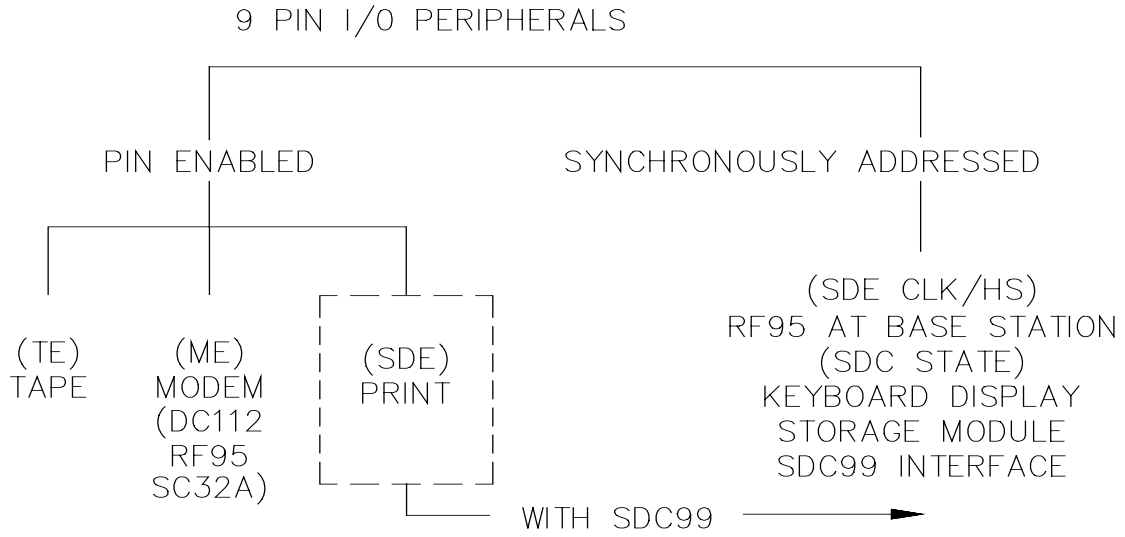


FIGURE 6.2-1. Hardware Enabled and Synchronously Addressed Peripherals

6.2 ENABLING AND ADDRESSING PERIPHERALS

While several peripherals may be connected in parallel to the 9-pin port, the CR10 has only one transmit line (pin 9) and one receive line (pin 4, Table 6.1-1). The CR10 selects a peripheral in one of two ways: 1) A specific pin is dedicated to that peripheral and the peripheral is enabled when the pin goes high; we will call this pin-enabled or simply enabled. 2) The peripheral is addressed; the address is sent on pin 9, each bit being synchronously clocked using pin 7. Pin 6 is set high while addressing.

6.2.1 PIN-ENABLED PERIPHERALS

Two pins are dedicated to specific devices, Tape Enable (pin 8) and Modem Enable (pin 5). Pin 6 (Synchronous Device Enable) can either be used as a Print Enable OR it can be used to address Synchronous Devices (Section 6.6).

Tape Enable (TE), pin 8, is raised to enable data transfer to tape. The SC92A Cassette Interface regulates 12 volts from the CR10 to 6V DC to power the RC35 recorder and also provides signal conditioning. ONLY ONE TAPE INTERFACE AND RECORDER MAY BE CONNECTED TO THE CR10.

Modem Enable (ME), pin 5, is raised to enable a modem that has raised the ring line. Modem/terminal peripherals include Campbell Scientific phone modems and computers or terminals using the SC32A RS232 interface.

The CR10 interprets a ring interrupt (Section 6.3) to come from a modem if the device raises the CR10's Ring line, and holds it high until the CR10 raises the ME line. Only one modem/ terminal may be connected to the CR10.

Print Peripherals are defined as peripherals which have an asynchronous serial communications port used to RECEIVE data transferred by the CR10. In most cases the print peripheral is a printer, but could also be an on-line computer or other device.

Synchronous Device Enable (SDE), pin 6, may be used to enable a print peripheral only when no other addressable peripherals are connected to the 9-pin connector. Use of the SDE line as an enable line maintains CR10 compatibility with printer-type peripherals which require a line to be held high (Data Terminal Ready) in order to receive data.

If output to both a print peripheral and an addressable peripheral is necessary the SDC99 Synchronous Device Interface is required. With the SDC99 the print peripheral functions as an addressable peripheral. If the SDC99 is not used, the print peripheral receives the address and data sent to the addressed peripheral. Synchronous addressing appears as garbage characters on a print peripheral.

6.2.2 ADDRESSED PERIPHERALS

The CR10 distinguishes itself from other Campbell Scientific dataloggers by the ability to address Synchronous Devices (SDs). SDs differ

from enabled peripherals in that they are not enabled solely by a hardware line (Section 6.2.1); an SD is enabled by an address synchronously clocked from the CR10 (Section 6.6).

Up to 16 SDs may be addressed by the CR10. Unlike an enabled peripheral, the CR10 establishes communication with an addressed peripheral before data are transferred. During data transfer an addressed peripheral uses pin 7 as a handshake line with the CR10.

Synchronously addressed peripherals include the CR10KD Keyboard Display, SM716 and SM192 Storage Modules, SDC99 Synchronous Device Interface (SDC99), and RF95 RF Modem when configured as a synchronous device. The SDC99 interface is used to address peripherals which are normally enabled (Figure 6.2-1).

6.3 RING INTERRUPTS

There are three peripherals that can raise the CR10's ring line; modems, the CR10KD Keyboard Display, and the RF Modem configured for synchronous device for communication (RF-SDC). The RF-SDC is used when the CR10 is installed at a telephone to RF base station.

When the Ring line is raised, the processor is interrupted, and the CR10 determines which peripheral raised the Ring line through a process of elimination (Figure 6.3-1). The CR10 raises the CLK/HS line forcing all SDs to drop the ring line. If the ring line is still high the peripheral is a modem, and the ME line is raised. If the ring line is low the CR10 addresses the Keyboard Display and RF-SDC to determine which device to service. (Section 6.6)

After the CR10 has determined which peripheral raised the Ring line, the hierarchy is as follows:

A modem which raises the Ring line will interrupt and gain control of the CR10. The one exception is that a modem cannot interrupt an active RF-SDC. A ring from a modem aborts data transfer to pin-enabled and addressed peripherals.

The CR10KD raises the ring line whenever a key is pressed. The CR10KD will not be serviced when the modem or RF-SDC is being serviced.

The ring from the CR10KD and RF-SDC is blocked when the SDE line is high, preventing it from interrupting data transfer to a pin-enabled print device.

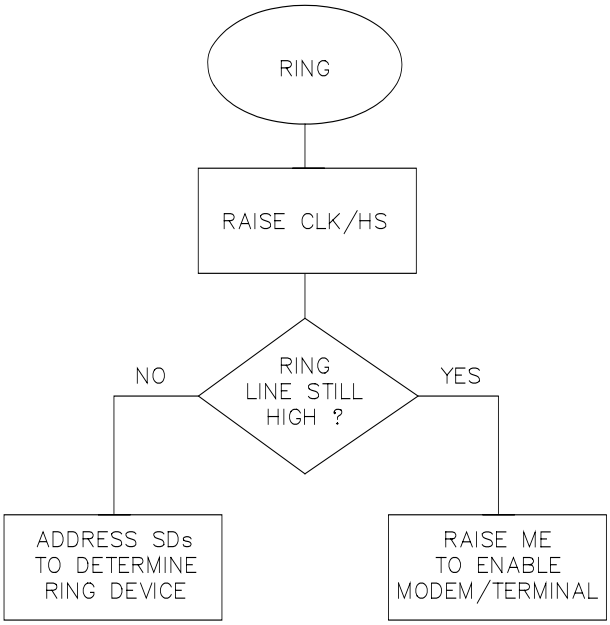


FIGURE 6.3-1. Servicing of Ring Interrupts

6.4 INTERRUPTS DURING DATA TRANSFER

Instruction 96 is used for on-line data transfer to peripherals (Section 4.1). Each peripheral connected to the CR10 requires an Instruction 96 with the appropriate parameter. If the CR10 is already communicating on the 9-pin connector when Instruction 96 is executed, the instruction puts the output request in a "queue" and program execution continues. As the 9-pin connector becomes available, each device in the queue will get its turn until the queue is empty.

Instruction 96 is aborted if a modem raises the Ring line. Data transfer to an addressed peripheral is aborted if the ring line is raised by a CR10KD or RF Modem configured as a synchronous device. Transfer of data is not resumed until the next time Instruction 96 is executed and the datalogger has exited telecommunications.

The *8 Mode is used to manually initiate data transfer from Final Storage to a peripheral. An abort flag is set if any key on the CR10KD or terminal is pressed during the transfer. Data transfer is stopped and the memory location displayed when the flag is set. During *8 data transfer the abort flag is checked as follows:

1. Comma delineated ASCII - after every 32 characters.
2. Printable ASCII - after every line.
3. Binary - after every 256 Final Storage locations.
4. Tape - after every block (512 Final Storage locations).

6.5 MODEM/TERMINAL PERIPHERALS

The CR10 considers any device with an asynchronous serial communications port which raises the Ring line (and holds it high until the ME line is raised) to be a modem peripheral. Modem/terminals include Campbell Scientific phone modems, and most computers, terminals, and modems using the SC32A Optically Isolated RS232 Interface.

When a modem raises the Ring line, the CR10 responds by raising the ME line. The CR10 must then receive carriage returns until it can establish baud rate. When the baud rate has been set, the CR10 sends a carriage return, line feed, "*".

The ME line is held high until the CR10 receives an "E" to exit telecommunications. The ME is also lowered if a character is not received after 40 seconds in the Telecommunications Command State (2 minutes in the Remote Keyboard State).

Some modems are quite noisy when not on line; it is possible for valid characters to appear in the noise pattern. For this reason, the CR10 counts all the invalid characters it receives from the time it answers a ring, and terminates communication (lowers the ME line and returns to the *0 Mode) after receiving 150 invalid characters.

6.6 SYNCHRONOUS DEVICE COMMUNICATION

The CR10 differs from other Campbell Scientific dataloggers by its ability to address Synchronous Devices (SDs). SDs differ from enabled peripherals (Section 6.2.1) in that they are not enabled solely by a hardware line. An SD is enabled by an address synchronously clocked from the CR10. Up to 16 SDs may be addressed by the CR10, requiring only three pins of the 9-pin connector.

Synchronous Device Communication (SDC) discussed here is for those peripherals which connect to the 9-pin serial port. This should not be confused with Synchronous Device for Measurement (SDM) peripherals connected to control ports 1, 2, and 3. (Although the communication protocol for SDMs is very similar, their addressing is independent of SDC addresses and they do not have a ring line.)

SD STATES

The CR10 and the SDs use a combination of the Ring, Clock Handshake (CLK/HS) and Synchronous Device Enable (SDE) lines to establish communication. The CR10 can put the SDs into one of six states.

STATE 1, the SD Reset State

The CR10 forces the SDs to the reset/request state by lowering the SDE and CLK/HS lines. The SD cannot drive the CLK/HS or RXD lines in State 1, however, it can raise the Ring line if service is needed. The SD can never pull the Ring low if a Modem/Terminal is holding it high. Data on TXD is ignored by the SD.

STATE 2, the SD Addressing State

The CR10 places the SDs in the addressing state by raising CLK/HS followed by or simultaneously raising SDE (Figure 6.6-1). TXD must be low while SDE and CLK/HS are changing to the high state.

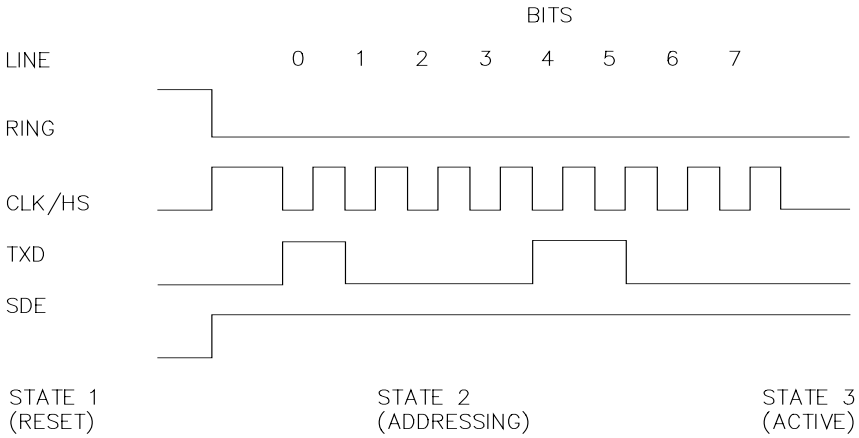


FIGURE 6.6-1. Addressing Sequence for the RF Modem

State 2 requires all SDs to drop the Ring line and prepare for addressing. The CR10 then synchronously clocks 8 bits onto TXD using CLK/HS as a clock. The least significant bit is transmitted first and is always logic high. Each bit transmitted is stable on the rising edge of CLK/HS. The SDs shift in bits from TXD on the rising edge of CLK/HS provided by the CR10. The CR10 can only address one device per State 2 cycle. More than one SD may respond to the address, however. State 2 ends when the 8th bit is received by the SD.

SDs implemented with shift registers decode the 4 most significant bits (bits 4, 5, 6, and 7) for an address. Bit 0 is always logic high. Bits 1, 2, and 3 are optional function selectors or commands. Addresses established to date are shown in Table 6.6-1 and are decoded with respect to the TXD line.

TABLE 6.6-1. SD Addresses

	B7	B6	B5	B4	B3	B2	B1	B0
SDC99 Printer	0	0	0	0	X	X	X	1
Storage Module	0	0	0	1	X	X	X	1
CR10 Keyboard	0	0	1	0	0	X	X	1
CR10 Display	0	0	1	0	1	X	X	1
Cr10 RF Modem	0	0	1	1	X	X	X	1
EPROM Storage Module	0	1	0	0	X	X	X	1

STATE 3, the SD Active State

The SD addressed by State 2, enters State 3. All other SDs enter State 4. An active SD returns to State 1 by resetting itself, or by the CR10 forcing it to reset.

Active SDs have different acknowledgment and communication protocols. Once addressed, the SD is free to use the CLK/HS, TXD, and RXD lines according to its protocol with the CR10. The CR10 may also pulse the SDE line after addressing, as long as the CLK/HS and SDE are not low at the same time.

STATE 4, the SD Inactive State

The SDs not addressed by State 2 enter State 4, if not able to reset themselves (e.g., SM192 Storage Module). Inactive SDs ignore data on the TXD line and are not allowed to use the CLK/HS or RXD lines. Inactive SDs may raise the Ring line to request service.

STATE 5

State 5 is a branch from State 1 when the SDE line is high and the CLK/HS line is low. The SDs must drop the Ring line in this state. This state is not used by SDs. The CR10 must force the SDs back to the reset state from State 5 before addressing SDs.

STATE 6

State 6 is a branch from State 1, like State 5, except the SDE line is low and the CLK/HS line is high. The SDs must drop the Ring line in this state.

6.7 MODEM/TERMINAL AND COMPUTER REQUIREMENTS

6.7.1 SC32A INTERFACE

The CR10 considers any device with an asynchronous serial communications port which raises the Ring line (and holds it high until the ME line is raised) to be a modem peripheral. Modems include Campbell Scientific phone modems, and most computers, terminals, and modems using the SC32A Optically Isolated RS232 Interface.

Most modem and print peripherals require the SC32A Optically Isolated RS232 Interface. The SC32A raises the CR10's ring line when it receives characters from a modem, and converts the CR10's logic levels (0 V logic low, 5V logic high) to RS232 logic levels.

The SC32A 25-pin port is configured as Data Communications Equipment (DCE) (see Table 6.7-1) for direct connection to Data Terminal Equipment (DTE), which includes most PCs and printers. For connection to DCE devices such as modems and some computers, a null modem cable is required.

When the SC32A receives a character from the terminal/computer (pin 2), 5 V is applied to the datalogger Ring line (pin 3) for one second or until the Modem Enable line (ME) goes high. The CR10 waits approximately 40 seconds to receive carriage returns, which it uses to establish baud rate. After the baud rate has been set the CR10 transmits a carriage return, line feed, "*", and enters the Telecommunica

tions Command State (Section 5). If the carriage returns are not received within the 40 seconds, the CR10 "hangs up".

TABLE 6.7-1. SC32A Pin Description

ABR = Abbreviation for the function name
PIN = Pin number
O = Signal Out of the SC32A to a peripheral
I = Signal Into the SC32A from peripheral

25-PIN FEMALE PORT:

PIN #	I/O	ABBREVIATION
1		GROUND
2	I	TX
3	O	RX
4	I	RTS (POWER)
5	O	CTS
6	O	DSR
7		GROUND
8	O	DCD
20	I	DTR (POWER)

9-PIN MALE PORT:

PIN #	ABBREVIATION
1	+5V INPUT
2	GROUND
3	RING
4	RX
5	ME
6	SDE
9	TX

NOTE: The SC32A has a jumper, which when used, passes data only when the ME line is high and the SDE line is low. The function of the jumper is to block data sent to SDs from being received by a computer/terminal used to initiate data transfer. Synchronous data will appear as garbage characters on a computer/terminal.

6.7.2 COMPUTER/TERMINAL REQUIREMENTS

Computer/terminal peripherals are usually configured as Data Terminal Equipment (DTE). Pins 4 and 20 are used as handshake lines, which are set high when the serial port is enabled. Power for the SC32A RS232 section is taken from these pins. For equipment configured as DTE (see Table 6.7-2) a direct ribbon cable connects the computer/terminal to the SC32A. Clear to Send (CTS) pin 5, Data Set Ready (DSR) pin 6, and Received Line Signal Detect (RLSD) pin 8 are held high by the SC32A (when the RS232 section is powered) which should satisfy hardware handshake requirements of the computer/terminal.

Table 6.7-2 lists the most common RS232 configuration for Data Terminal Equipment.

TABLE 6.7-2. DTE Pin Configuration

PIN = 25-pin connector number
ABR = Abbreviation for the function name
O = Signal Out of terminal to another device
I = Signal Into terminal from another device

PIN	ABR	I/O	FUNCTION
2	TD	O	Transmitted Data: Data is transmitted from the terminal on this line.
3	RD	I	Received Data: Data is received by the terminal on this line.
4	RTS	O	Request to Send: The terminal raises this line to ask a receiving device if the terminal can transmit data.
5	CTS	I	Clear to Send: The receiving device raises this line to let the terminal know that the receiving device is ready to accept data.
20	DTR	O	Data Terminal Ready: The terminal raises this line to tell the modem to connect itself to the telephone line.
6	DSR	I	Data Set Ready: The modem raises this line to tell the terminal that the modem is connected to the phone line.
8	DCD	I	Data Carrier Detect: The modem raises this line to tell the terminal that the modem is receiving a valid carrier signal from the phone line.

22	RI	I	Ring Indicator: The modem raises this line to tell the terminal that the phone is ringing.	7	SG	Signal Ground: Voltages are measured relative to this point.

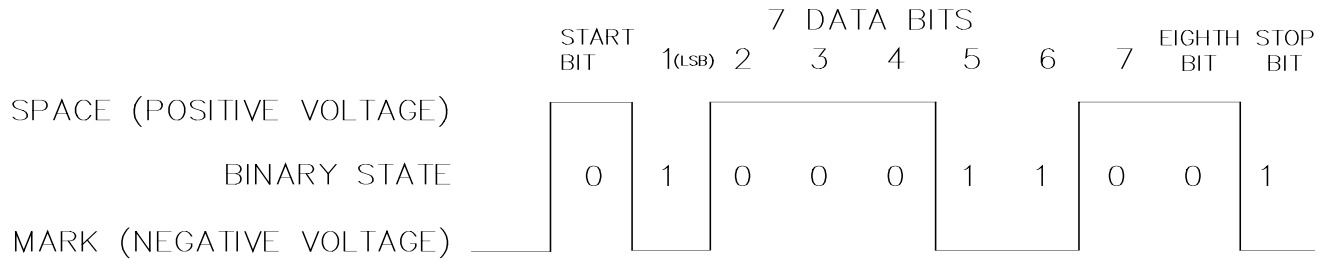


FIGURE 6.7-1. Transmitting the ASCII Character 1

If the computer/terminal is configured as DCE equipment (pin 2 is an input for RD), a null modem cable is required. See the SC32A manual for details.

6.7.3 COMMUNICATION PROTOCOL/TROUBLE SHOOTING

The ASCII standard defines an alphabet consisting of 128 different characters where each character corresponds to a number, letter, symbol, or control code.

An ASCII character is a binary digital code composed of a combination of seven "bits", each bit having a binary state of 1 (one) or 0 (zero). For example, the binary equivalent for the ASCII character "1" is 0110001 (decimal 49).

ASCII characters are transmitted one bit at a time, starting with the 1st (least significant) bit. During data transmission the marking condition is used to denote the binary state 1, and the spacing condition for the binary state 0. The signal is considered marking when the voltage is more negative than minus three volts with respect to ground, and spacing when the voltage is more positive than plus three volts.

Most computers use 8-bits (1 byte) for data communications. The 8th bit is sometimes used for a type of error checking called parity-checking. Even parity binary characters have an even number of 1's, odd-parity characters have an odd number of 1's. When parity checking is used, the 8th bit is set to either a 1 or a 0 to make the parity of the character correct. The CR10 ignores the 8th bit of a character that is receives, and transmits the 8th bit as a binary 0. This method is generally described as "no parity".

To separate ASCII characters a Start bit is sent before the 1st bit, and a Stop bit is sent after the 8th bit. The start bit is always a space, and the stop bit is always a mark. Between characters the signal is in the marking condition.

Figure 6.7-1 shows how the ASCII character "1" is transmitted. When transmitted by the CR10 using the SC32A RS232 interface spacing and marking voltages are positive and negative, as shown. Signal voltages at the CR10 I/O port are 5V in the spacing condition, and 0V in the marking condition.

BAUD RATE

BAUD RATE is the number of bits transmitted per second. The CR10 can communicate at 300, 1200, 9600, and 76,800 baud. In the Telecommunications State, the CR10 will set its baud rate to match the baud rate of the computer/terminal.

Typically the baud rate of the modem/computer/terminal is set either with dip switches, or programmed from the keyboard. The instrument's instruction manual should explain how to set it.

DUPLEX

Full duplex means that two devices can communicate in both directions simultaneously. Half duplex means that the two devices must send and receive alternately. Full duplex should always be specified when communicating with Campbell Scientific peripherals and modems. However, communication between some Campbell Scientific modems (such as the RF95 RF modem) is carried out in a half duplex fashion. This can affect the way commands should be sent to and received from such a modem, especially when implemented by computer software.

To overcome the limitations of half duplex, some communications links expect a terminal sending data to also write the data to the screen. This saves the remote device having to echo that data back. If, when communicating with a Campbell Scientific device, characters are displayed twice (in pairs), it is likely that the terminal is set to half duplex rather than the correct setting of full duplex.

IF NOTHING HAPPENS

If the CR10 is connected to the SC32A RS232 interface and a modem/terminal, and an "*" is not received after sending carriage returns:

1. Verify that the CR10 has power AT THE 12V AND GROUND INPUTS, and that the cables connecting the devices are securely connected.

2. Verify that the port of the modem/terminal is an asynchronous serial communications port configured as DTE (see Table 6.7-2). The most common problems occur when the user tries to use a parallel port, or doesn't know the port assignments, i.e. COM1 or COM2. IBM, and most compatibles come with a Diagnostic disk which can be used to identify ports, and their assignments. If the serial port is standard equipment, then the operators manual should give you this information.

Some serial ports, e.g., the Super Serial Card for Apple computers, can be configured as DTE or DCE with a jumper block. Pin functions must match with Table 6.7-2.

If you are using a computer to communicate with the datalogger, communication software must be used to enable the serial port and to make the computer function as a terminal. Campbell Scientific's TERM program (part of the PC208 Datalogger Support Software) provides this function for IBM PC/XT/AT/PS-2's and compatibles. The port should be enabled for 300, 1200, or 9600 baud, 8 data bits, 1 stop bit, and no parity.

If you are not sure that your computer/terminal is sending or receiving characters, there is a simple way to verify it. Make sure that the duplex is set to full. Next, take a paper clip and connect one end to pin 2, and the other end to pin 3 of the serial port. Each character typed on the keyboard will be displayed only if transmitted from the terminal on pin 2, and received on pin 3 (if duplex is set to half, the character will be displayed once if it is not transmitted, or twice if it is transmitted).

IF GARBAGE APPEARS

If garbage characters appear on the display, check that the baud rate is supported by the CR10. If the baud rate is correct, verify that the computer/terminal is set for 8 data bits, and no parity. Garbage will appear if 7 data bits and no parity are used. If the computer/terminal is set to 8 data bits and even or odd parity, communication cannot be established.

SECTION 7. MEASUREMENT PROGRAMMING EXAMPLES

This section gives some examples of Input Programming for common sensors used with the CR10. These examples detail only the connections, Input, Program Control, and Processing Instructions necessary to perform measurements and store the data in engineering units in Input Storage. Output Processing Instructions are omitted (see Section 8 for some processing and program control examples). It is left to the user to program the necessary instructions to obtain the final data in the form desired. No output to final storage will take place without additional programming.

The examples given in this section would likely be only fragments of larger programs. In general, the examples are written with the measurements made by the lowest numbered channels, the instructions at the beginning of the program table, and low number Input Storage Locations used to store the data. It is unlikely that an application and CR10 configuration exactly duplicates that assumed in an example.

These examples are not meant to be used verbatim; sensor calibration, input channels, and input locations must be adjusted for the actual circumstances. Unless otherwise noted, all excitation channels are switched analog output.

7.1 SINGLE-ENDED VOLTAGE - LI200S SILICON PYRANOMETER

The silicon pyranometer outputs a current which is dependent upon the solar radiation incident upon the sensor. The current is measured as the voltage drop across a fixed resistor. The Campbell Scientific LI200S uses a 100 ohm resistor. The calibration supplied by LI-COR, the manufacturers of the pyranometer, is given in µA/kW/m^2. To convert calibration values to volts multiply the µA calibration by the resistance of the fixed resistor.

The calibration of the pyranometer used in this example is 76.9 µA/kW/m^2, which when multiplied by 100 ohms equals 7.69mV/kW/m^2. The multiplier used to convert the voltage reading to kW/m^2 is 1/7.69mV/kW/m^2 = 0.13004 kW/m^2/mV.

Most LI-COR calibrations run between 60 and 90 µA/kW/m^2, which correspond to calibrations of 6.0 to 9.0mV/kW/m^2. Above the earth's atmosphere, the flux density through a surface normal to the solar beam is 1.36kW/m^2; radiation at the surface of the earth will be less than this. Thus, the 25mV scale provides an adequate range (9.0mV/kW/m^2 x 1.36 kW/m^2 < 25mV).

CONNECTIONS

The pyranometer output can be measured with a single-ended voltage measurement on channel 5. There are twice as many single-ended channels as differential channels and they are numbered accordingly: Single-ended channel 5 is the high side of differential channel 3 (3H); the low side (3L) is single-ended channel 6.

PROGRAM

```
01:      P1          Volt (SE)
  01:     1          Rep
  02:    23          25 mV 60 Hz rejection
                     Range
  03:     5          IN Chan
  04:     1          Loc [:RAD kW/m2]
  05:      .13004    Mult
  06:     0          Offset
```

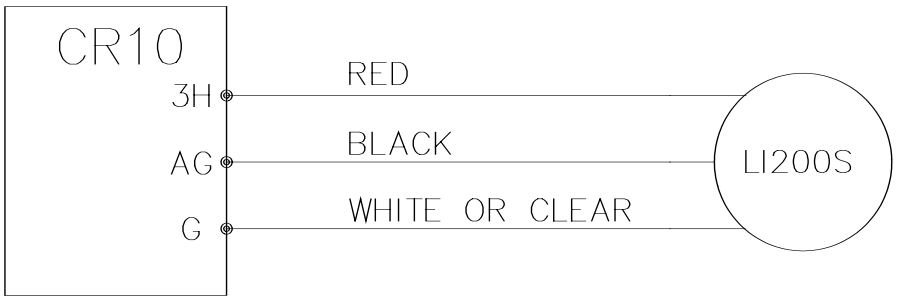


FIGURE 7.1-1. Wiring Diagram for LI200S

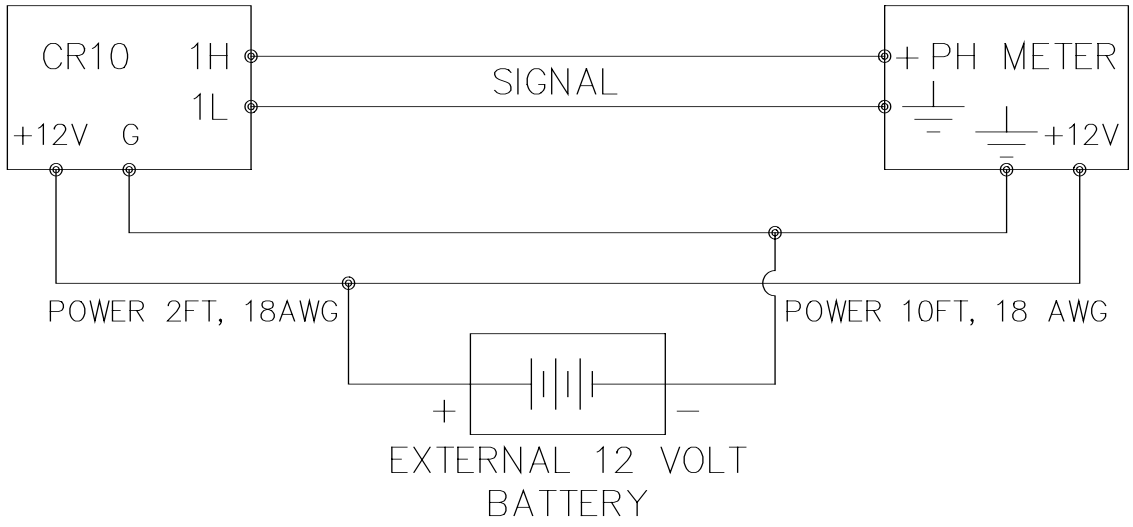


FIGURE 7.2-1. Typical Connection for Active Sensor with External Battery

7.2 DIFFERENTIAL VOLTAGE MEASUREMENT

Some sensors either contain or require active signal conditioning circuitry to provide an easily measured analog voltage output. Generally, the output is referenced to the sensor ground. The associated current drain usually requires a power source external to the CR10. A typical connection scheme where AC power is not available and both the CR10 and sensor are powered by an external battery is shown in Figure 7.2-1. Since a single-ended measurement is referenced to the CR10 ground, any voltage difference between the sensor ground and CR10 ground becomes a measurement error. A differential measurement avoids this error by measuring the signal between the 2 leads without reference to ground. This example analyzes the potential error on a water pH measurement using a Martek Mark V water quality analyzer.

The wire used to supply power from the external battery is 18 AWG with an average resistance of 6.5 ohms/1000 ft. The power leads to the CR10 and pH meter are 2 ft and 10 ft, respectively. Typical current drain for the pH meter is 300 mA. When making measurements, the CR10 draws about 35 mA. Since voltage is equal to current multiplied by resistance (V=IR), ground voltages at the pH meter and the CR10 relative to battery ground are:

pH meter ground =
0.3A x 10/1000 x 6.5ohms = +0.0195V

CR10 ground =
0.035A x 2/1000 x 6.5ohms = +0.0005V

Ground at the pH meter is 0.0190 V higher than ground at the CR10. The meter output is 0-1 volt referenced to meter ground, for the full range of 14 pH units, or 0.0714 V/pH. Thus, if the output is measured with a single-ended voltage measurement, it is 0.0190 V or 0.266 pH units too high. If this offset remained constant, it could be corrected in programming. However, it is better to use a differential voltage measurement which does not rely on the current drain remaining constant. The program that follows illustrates the use of Instruction 2 to make the measurement. A multiplier of 0.014 is used to convert the millivolt output into pH units.

PROGRAM

```
01:      P2          Volt (DIFF)
  01:      1         Rep
  02:     25         2500 mV 60 Hz rejection
  03:      1         IN Chan
  04:      1         Loc [:pH      ]
  05:      0.014     Mult
  06:      0         Offset
```

FIGURE 7.3-1. CR10TCR Mounted on the CR10 Wiring Panel

7.3 THERMOCOUPLE TEMPERATURES USING THE OPTIONAL CR10TCR TO MEASURE THE REFERENCE TEMPERATURE

The CR10TCR Thermocouple Reference is a temperature reference for thermocouples measured with the CR10 Measurement and Control Module. When installed, the CR10TCR lies between the two analog input terminal strips of the CR10 Wiring Panel (see Figure 7.3-1). The CR10TCR circuitry, measurement, and specifications are equivalent to Campbell Scientific's 107 Temperature Probe.

The CR10TCR is connected to single-ended channel 1 (1H), excitation channel 3 (E3) and analog ground (AG). It is measured with Instruction 11 which excites the probe with a 2.5VAC excitation, makes a single ended measurement and calculates temperature (°C). Five differential thermocouples are measured with Instruction 14. (Refer to the CR10TCR Manual for instructions on measuring a thermocouple in differential channel 1.)

The temperature (°C) of the CR10TCR is stored in Input Location 1 and the thermocouple temperatures (°C) in Locations 2-6.

PROGRAM

```
01:     P11     Temp 107 Probe
  01:     1     Rep
  02:     1     IN Chan
  03:     3     Excite all reps w/EXchan 3
  04:     1     Loc [:REF TEMP ]
  05:     1     Mult
  06:     0     Offset

02:     P14     Thermocouple Temp (DIFF)
  01:     5     Reps
  02:    22     7.5 mV 60 Hz rejection Range

  03:     2     IN Chan
  04:     1     Type T (Copper-Constantan)
  05:     1     Ref Temp Loc REF TEMP
  06:     2     Loc [:TC #1    ]
  07:     1     Mult
  08:     0     Offset
```

7.4 THERMOCOUPLE TEMPERATURES USING AN EXTERNAL REFERENCE JUNCTION

When a number of thermocouple measurements are made at some distance from the CR10, it is often better to use a reference junction box located at the site rather than use the CR10TCR Thermocouple Reference. Use of the external reference junction reduces the required length of expensive thermocouple wire as regular copper wire can be used between the junction box (J-box) and CR10. In addition, if the temperature gradient between the J-box and the thermocouple measurement junction is smaller than the gradient between the CR10 and the measurement junction, thermocouple accuracy is improved.

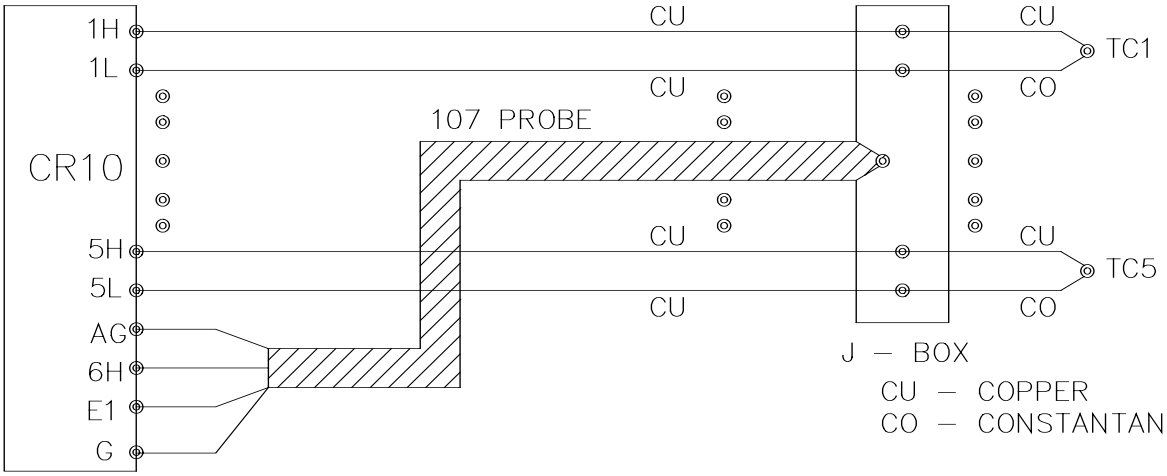


FIGURE 7.4-1. Thermocouples with External Reference Junction

In the following example, an external temperature measurement is used as the reference for 5 thermocouple measurements. A Campbell Scientific 107 Temperature Probe is used to measure the reference temperature. The connection scheme is shown in Figure 7.4-1.

If a more accurate reference temperature is needed, Campbell Scientific's TCR6 utilizes a 100 ohm PRT to measure the reference temperature and provides better insulation for a more isothermal reference.

The temperature (°C) of the 107 Probe is stored in Input Location 1 and the thermocouple temperatures (°C) in Locations 2-6.

PROGRAM

```
01:     P11         Temp 107 Probe
  01:     1         Rep
  02:    11         IN Chan
  03:     1         Excite all reps w/EXchan 1
  04:     1         Loc [:REF TEMP ]
  05:     1         Mult
  06:     0         Offset

02:     P14         Thermocouple Temp (DIFF)
  01:     5         Reps
  02:    22         7.5 mV 60 Hz rejection Range

  03:     1         IN Chan
  04:     1         Type T (Copper-
                    Constantan)
  05:     1         Ref Temp Loc REF TEMP
  06:     2         Loc [:TC #1    ]
  07:     1         Mult
  08:     0         Offset
```

7.5 107 TEMPERATURE PROBE

Instruction 11 excites Campbell Scientific's 107 Thermistor Probe (or the thermistor portion of the 207 temperature and relative humidity probe) with a 2 VAC excitation, makes a single ended measurement and calculates temperature (°C) with a fifth order polynomial. In this example, the temperatures are obtained from three 107 probes. The measurements are made on single-ended channels 1-3 and the temperatures are stored in Input Locations 1-3.

CONNECTIONS

The black leads from the probes go to excitation channel 1, the purple leads go to analog ground (AG), the clear leads go to ground (G), and the red leads go to single-ended channels 1, 2, and 3 (channel 1H, channel 1L, and channel 2H, respectively).

PROGRAM

```
01:     P11         Temp 107 Probe
  01:     3         Reps
  02:     1         IN Chan
  03:     1         Excite all reps w/EXchan 1
  04:     1         Loc [:107 T #1 ]
  05:     1         Mult
  06:     0         Offset
```

7.6 207 TEMPERATURE AND RH PROBE

Instruction 12 excites and measures the RH portion of the Campbell Scientific 207 Temperature and Relative Humidity probe. This instruction relies on a previously measured temperature (°C) to compute RH from the probe resistance. Instruction 11 is used to obtain the

temperatures of the three probes which are stored in Input Locations 1-3; the RH values are stored in Input Locations 4-6. The temperature measurements are made on single-ended input channels 1-3, just as in example 7.5. The program listed below is a continuation of the program given in example 7.5.

CONNECTIONS

The black leads from the probes are connected to excitation channel 1, the purple leads are connected to analog ground (AG), and the clear leads are connected to Ground (G). The red leads are from the thermistor circuit and are connected to single-ended channels 1-3 (1H, 1L, 2H). The white leads are from the RH circuit and are connected to single-ended channels 4-6 (2L, 3H, and 3L). The correct order must be maintained when connecting the red and white leads; i.e., the red lead from the first probe is connected to single-ended channel 1H and the white lead from that probe is connected to single-ended channel 2L, etc.

PROGRAM *(continuation of previous example)*

```
02:     P12         RH 207 Probe
  01:     3         Reps
  02:     4         IN Chan
  03:     1         Excite all reps w/EXchan 1
  04:     1         Temperature Loc 107 T #1
  05:     4         Loc [:RH #1    ]
  06:     1         Mult
  07:     0         Offset
```

7.7 ANEMOMETER WITH PHOTOCHOPPER OUTPUT

An anemometer with a photochopper transducer produces a pulse output which is measured by the CR10's Pulse Count Instruction. The Pulse Count Instruction with a Configuration Code of 20, measures "high frequency pulses", "discards data from excessive intervals", and "outputs the reading as a frequency" (Hz = pulses per second). The frequency output is the only output option that is independent of the scan rate.

The anemometer used in this example is the R. M. Young Model 12102D Cup Anemometer, with a 10 window chopper wheel. The photochopper circuitry is powered from the CR10 12 V supply; AC power or back-up batteries should be used to compensate for the increased current drain.

Wind speed is desired in meters per second (m/s). There is a pulse each time a window in the chopper wheel, which revolves with the cups, allows light to pass from the source to the photoreceptor. Because there are 10 windows in the chopper wheel, there are 10 pulses per revolution. Thus, 1 revolution per minute (rpm) is equal to 10 pulses per 60 seconds (1 minute) or 6 rpm = 1 pulse per second (Hz). The manufacturer's calibration for relating wind speed to rpm is:

$$\text{Wind(m/s)} = (0.01632 \text{ m/s})/\text{rpm} \times \text{Xrpm} + 0.2 \text{ m/s}$$

The result of the Pulse Count Instruction (Configuration Code = 20) is X pulses per sec. (Hz). The multiplier and offset to convert XHz to meters per second are: Wind (m/s) = (0.01632 m/s)/rpm x (6 rpm/Hz) x XHz + 0.2 m/s

$$\text{Wind (m/s)} = (0.09792 \text{ m/s})/\text{Hz} \times \text{XHz} + 0.2 \text{ m/s}$$

PROGRAM

```
01:     P3          Pulse
  01:     1         Rep
  02:     1         Pulse Input Chan
  03:    20         High frequency; Output Hz.

  04:    10         Loc [:WS MPH   ]
  05:     .09792    Mult
  06:     0.2       Offset
```

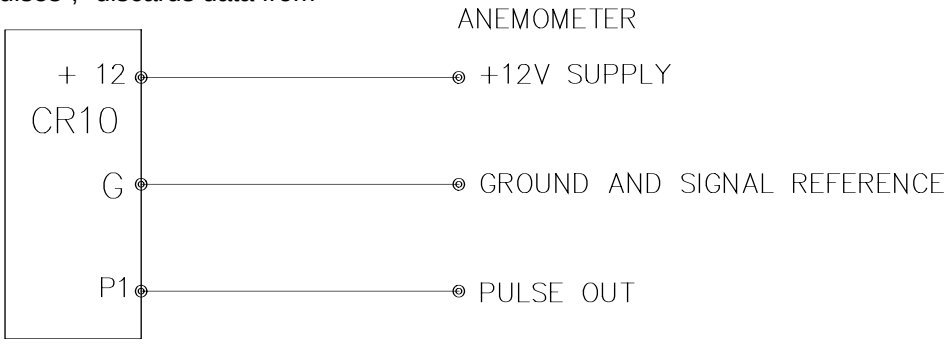


FIGURE 7.7-1. Wiring Diagram for Anemometer

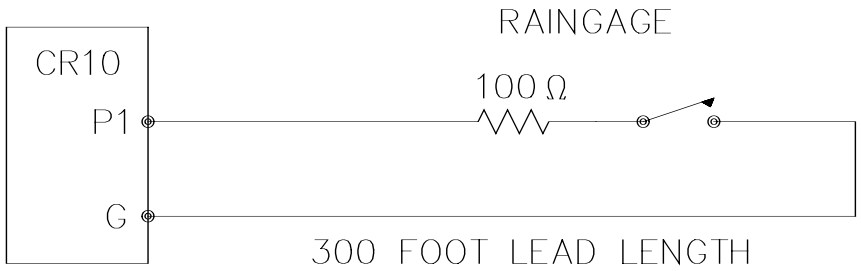


FIGURE 7.8-1. Wiring Diagram for Rain Gage with Long Leads

7.8 TIPPING BUCKET RAIN GAGE WITH LONG LEADS

A tipping bucket rain gage is measured with the Pulse Count Instruction configured for Switch Closure. Counts from long intervals will be used, as the final output desired is total rainfall (obtained with Instruction 72, Totalize). If counts from long intervals were discarded, less rainfall would be recorded than was actually measured by the gage (assuming there were counts in the long intervals). Output is desired in millimeters of precipitation. The gage is calibrated for a 0.01 inch tip, therefore, a multiplier of 0.254 is used.

In a long cable there is appreciable capacitance between the lines. The capacitance is discharged across the switch when it closes. In addition to shortening switch life, a transient may be induced in other wires packaged with the rain gage leads each time the switch closes. The 100 ohm resistor protects the switch from arcing and the associated transient from occurring, and should be included any time leads longer than 100 feet are used with a switch closure.

PROGRAM

```
01:      P3          Pulse
  01:      1         Rep
  02:      1         Pulse Input Chan
  03:      2         Switch closure
  04:     11         Loc [:RAIN mm  ]
  05:      0.254     Mult
  06:      0         Offset
```

7.9 100 OHM PRT IN 4 WIRE HALF BRIDGE

Instruction 9 is the best choice for accuracy where the Platinum Resistance Thermometer (PRT) is separated from other bridge completion resistors by a lead length having more than a few thousandths of an ohm resistance. In this example, it is desired to measure a temperature in the range of -10 to 40°C. The length of the cable from the CR10 to the PRT is 500 feet.

Figure 7.9-1 shows the circuit used to measure the PRT. The 10 kohm resistor allows the use of a high excitation voltage and low voltage ranges on the measurements. This insures that noise in the excitation does not have an effect on signal noise. Because the fixed resistor (R_f) and the PRT (R_s) have approximately the same resistance, the differential measurement of the voltage drop across the PRT can be made on the same range as the differential measurement of the voltage drop across R_f.

If the voltage drop across the PRT (V_2) is kept under 50mV, self heating of the PRT should be less than 0.001°C in still air. The best resolution is obtained when the excitation voltage is large enough to cause the signal voltage to fill the measurement voltage range. The resolution of this measurement on the 25mV range is +0.04°C. The voltage drop across the PRT is equal to V_x multiplied by the ratio of R_s to the total resistance, and is greatest when R_s is greatest (R_s=115.54 ohms at 40°C). To find the maximum excitation voltage that can be used, we assume V_2 equal to 25 mV and use Ohm's Law to solve for the resulting current, I.

$I = 25mV/R_s = 25mV/115.54 \text{ ohms} = 0.216 \text{ mA}$

Next solve for V_x:

$$V_x = I(R_1+R_s+R_f) = 2.21V$$

If the actual resistances were the nominal values, the CR10 would not over range with V_x = 2.2 V. To allow for the tolerances in the actual resistances, it is decided to set V_x equal to 2.1 volts (e.g., if the 10 kohms resistor is 5% low, then $R_s/(R_1+R_s+R_f)$=115.54/9715.54, and V_x must be 2.102V to keep V_s less than 25mV).

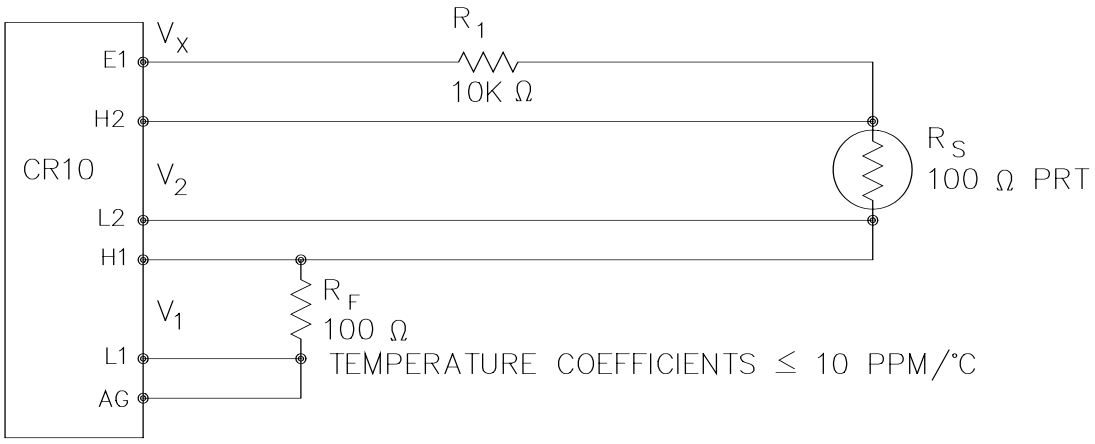


FIGURE 7.9-1. Wiring Diagram for PRT in 4 Wire Half Bridge

The result of Instruction 9 when the first differential measurement (V_1) is not made on the 2.5 V range is equivalent to R_s/R_f. Instruction 16 computes the temperature (°C) for a DIN 43760 standard PRT from the ratio of the PRT resistance at the temperature being measured to its resistance at 0°C (R_s/R_0). Thus, a multiplier of R_f/R_0 is used in Instruction 9 to obtain the desired intermediate, R_s/R_0 ($=R_s/R_f \times R_f/R_o$). If R_s and R_0 were each exactly 100 ohms, the multiplier would be 1. However, neither resistance is likely to be exact. The correct multiplier is found by connecting the PRT to the CR10 and entering Instruction 9 with a multiplier of 1. The PRT is then placed in an ice bath (@ 0°C; $R_s=R_0$), and the result of the bridge measurement is read using the *6 Mode. The reading is R_s/R_f, which is equal to R_o/R_f since $R_s=R_o$. The correct value of the multiplier, R_f/R_0, is the reciprocal of this reading. The initial reading assumed for this example was 0.9890. The correct multiplier is: $R_f/R_0 = 1/0.9890 = 1.0111$.

The fixed 100 ohm resistor must be thermally stable. Its precision is not important because the exact resistance is incorporated, along with that of the PRT, into the calibrated multiplier.

The 10 ppm/°C temperature coefficient of the fixed resistor will limit the error due to its change in resistance with temperature to less than 0.15°C over the specified temperature range. Because the measurement is ratiometric (R_s/R_f), the properties of the 10 kohm resistor do not affect the result.

PROGRAM

```
01:      P9          Full BR w/Compensation
  01:      1         Rep
  02:     23         25 mV 60 Hz rejection EX
                     Range
  03:     23         25 mV 60 Hz rejection BR
                     Range
  04:      1         IN Chan
  05:      1         Excite all reps w/EXchan 1
  06:   2100         mV Excitation
  07:      1         Loc [:Rs/Ro    ]
  08:      1.0111    Mult (Rf/Ro)
  09:      0         Offset

02:      P16         Temperature RTD
  01:      1         Rep
  02:      1         R/Ro Loc Rs/Ro
  03:      2         Loc [:TEMP C   ]
  04:      1         Mult
  05:      0         Offset
```

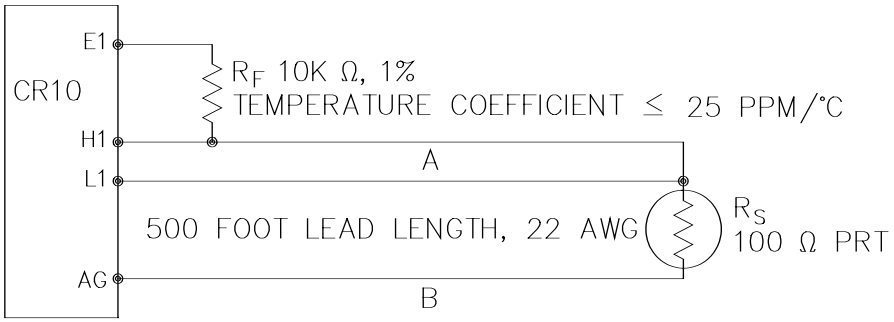


FIGURE 7.10-1. 3 Wire Half Bridge Used to Measure 100 ohm PRT

7.10 100 OHM PRT IN 3 WIRE HALF BRIDGE

The temperature measurement requirements in this example are the same as in Section 7.9. In this case, a three wire half bridge, Instruction 7, is used to measure the resistance of the PRT. The diagram of the PRT circuit is shown in Fig. 7.10-1.

As in the example in Section 7.9, the excitation voltage is calculated to be the maximum possible, yet allow the +25 mV measurement range. The 10 kohm resistor has a tolerance of ±1%; thus, the lowest resistance to expect from it is 9.9 kohms. We calculate the maximum excitation voltage (V_x) to keep the voltage drop across the PRT less than 25 mV:

$$0.025V > V_x\ 115.54/(9900+115.54);$$
$$V_x < 2.17\ V$$

The excitation voltage used is 2.1 V.

The multiplier used in Instruction 7 is determined in the same manner as in Section 7.9. In this example, the multiplier (R_f/R_0) is assumed to be 100.93.

The 3 wire half bridge compensates for lead wire resistance by assuming that the resistance of wire A is the same as the resistance of wire B. The maximum difference expected in wire resistance is 2%, but is more likely to be on the order of 1%. The resistance of R_S calculated with Instruction 7, is actually R_S plus the difference in resistance of wires A and B. The average resistance of 22 AWG wire is 16.5 ohms per 1000 feet, which would give each 500 foot lead wire a nominal resistance of 8.3 ohms. Two percent of 8.3 ohms is 0.17 ohms. Assuming that the greater resistance is in wire B, the resistance measured for the PRT (R_0 = 100 ohms) in the ice bath would be 100.17 ohms, and the resistance at 40°C would be 115.71. The measured ratio R_S/R_0 is 1.1551; the actual ratio is 115.54/100 = 1.1554. The temperature computed by Instruction 16 from the measured ratio would be about 0.1°C lower than the actual temperature of the PRT. This source of error does not exist in the example in Section 7.9, where a 4 wire half bridge is used to measure PRT resistance.

The advantages of the 3 wire half bridge are that it only requires 3 lead wires going to the sensor and takes 2 single-ended input channels, whereas the 4 wire half bridge requires 4 wires and 2 differential channels.

PROGRAM

```
01:      P7        3 Wire Half Bridge
  01:     1        Rep
  02:    23        25 mV 60 Hz rejection Range
  03:     1        IN Chan
  04:     1        Excite all reps w/EXchan 1
  05:  2100        mV Excitation
  06:     1        Loc [:Rs/Ro    ]
  07:  100.93      Mult
  08:     0        Offset

02:      P16       Temperature RTD
  01:     1        Rep
  02:     1        R/Ro Loc Rs/Ro
  03:     2        Loc [:TEMP C   ]
  04:     1        Mult
  05:     0        Offset
```

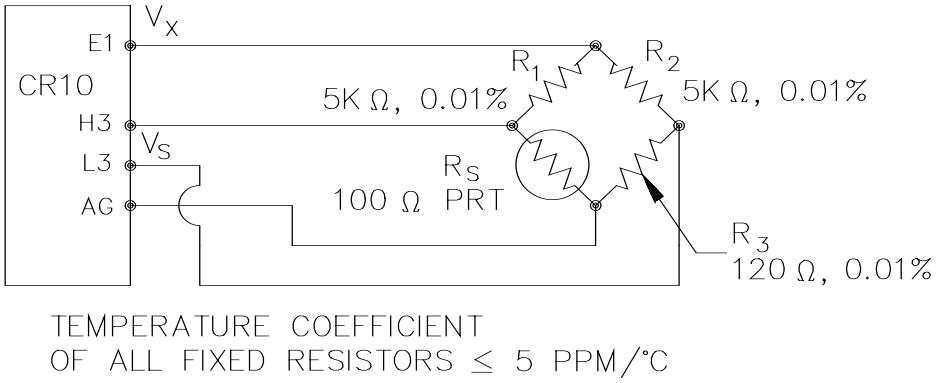


FIGURE 7.11-1. Full Bridge Schematic for 100 ohm PRT

7.11 100 OHM PRT IN 4 WIRE FULL BRIDGE

This example describes obtaining the temperature from a 100 ohm PRT in a 4 wire full bridge (Instruction 6). The temperature being measured is in a constant temperature bath and is to be used as the input for a control algorithm. The PRT in this case does not adhere to the DIN standard (alpha = 0.00385) used in the temperature calculating Instruction 16. Alpha is defined as $((R_{100}/R_0)-1)/100$, where R_{100} and R_0 are the resistances of the PRT at 100°C and 0°C, respectively. In this PRT alpha is equal to 0.00392.

The result given by Instruction 6 (X) is 1000 V_s/V_x (where V_s is the measured bridge output voltage, and V_x is the excitation voltage) which is:

$$X = 1000\ (R_s/(R_s+R_1)-R_3/(R_2+R_3))$$

The resistance of the PRT (R_s) is calculated with the Bridge Transform Instruction 59:

$$R_s = R_1\ X'/(1-X')$$

Where

$$X' = X/1000 + R_3/(R_2+R_3)$$

Thus, to obtain the value R_s/R_0, ($R_0 = R_s$ @ 0°C) for the temperature calculating Instruction 16, the multiplier and offset used in Instruction 6 are 0.001 and $R_3/(R_2+R_3)$, respectively. The multiplier used in Instruction 59 to obtain R_s/R_0 is R_1/R_0 (5000/100 = 50).

It is desired to control the temperature bath at 50°C with as little variation as possible. High resolution is needed so that the control algorithm will be able to respond to minute changes in temperature. The highest resolution is obtained when the temperature range results in an output voltage (V_s) range which fills the measurement range selected in Instruction 6. The full bridge configuration allows the bridge to be balanced (V_s = 0V) at or near the control temperature. Thus, the output voltage can go both positive and negative as the bath temperature changes, allowing the full use of the measurement range.

The resistance of the PRT is approximately 119.7 ohms at 50°C. The 120 ohm fixed resistor balances the bridge at approximately 51°C. The output voltage is:

$$V_s = V_x\ [R_s/(R_s+R_1) - R_3/(R_2+R_3)]$$

$$= Vx\ [R_s/(R_s+5000) - 0.023438]$$

The temperature range to be covered is ±50 ±10°C. At 40°C R_s is approximately 115.8 ohms, or:

$$V_s = -802.24x10\text{-}6\ V_x$$

Even with an excitation voltage (V_x) equal to 2500 mV, V_s can be measured on the +2.5 mV scale (40°C = 115.8 ohms = -2.006 mV, 60°C = 123.6 ohms = 1.714 mV). There is a change of approximately 2 mV from the output at 40°C to the output at 51°C, or 181 µV/°C. With a resolution of 0.33 µV on the 2.5 mV range, this means that the temperature resolution is 0.0018°C.

The 5 ppm per °C temperature coefficient of the fixed resistors was chosen so that their 0.01% accuracy tolerance would hold over the desired temperature range.

The relationship between temperature and PRT resistance is slightly nonlinear one. Instruction 16 computes this relationship for a DIN standard PRT where the nominal temperature

coefficient is 0.00385/°C. The change in nonlinearity of a PRT with the temperature coefficient of 0.00392/°C is minute compared with the slope change. Entering a slope correction factor of 0.00385/0.00392 = 0.98214 as the multiplier in Instruction 16 results in a calculated temperature which is well within the accuracy specifications of the PRT.

PROGRAM

```
01:      P6          Full Bridge
  01:     1          Rep
  02:    21          2.5 mV 60 Hz rejection Range

  03:     3          IN Chan
  04:     1          Excite all reps w/EXchan 1
  05:  2500          mV Excitation
  06:    11          Loc [:Rs/Ro    ]
  07:     0.001      Mult
  08:      .02344    Offset

02:      P59         BR Transform Rf[X/(1-X)]
  01:     1          Rep
  02:    11          Loc [:Rs/Ro    ]
  03:    50          Multiplier (Rf) =R1/R0

03:      P16         Temperature RTD
  01:     1          Rep
  02:    11          R/Ro Loc Rs/Ro
  03:    12          Loc [:TEMP C   ]
  04:      .98214    Mult
  05:     0          Offset
```

7.12 PRESSURE TRANSDUCER - 4 WIRE FULL BRIDGE

This example describes a measurement made with a Druck PDCR 930 depth measurement pressure transducer. The pressure transducer was ordered for use with 5 volt positive or negative excitation and has a range of 5 psi or about 3.5 meters of water. The transducer is used to measure the depth of water in a stilling well.

Instruction 6, 4 Wire Full Bridge, is used to measure the pressure transducer. The high output of the semiconductor strain gage necessitates the use of the 25 mV input range. The sensor is calibrated by connecting it to the CR10 and using Instruction 6, an excitation voltage of 2500 mV, a multiplier of 1 and an offset of 0, noting the readings (*6 Mode) with 10 cm of water above the sensor and with 334.6 cm of water above the sensor. The output of Instruction 6 is 1000 V_S/V_X or millivolts per volt excitation. At 10 cm the reading is 0.19963 mV/V and at 334.6 cm the reading is 6.6485 mV/V. The multiplier to yield output in cm is:

(334.6 - 10)/(6.6485-.19963) =
50.334 cm/mV/V

The offset is determined after the pressure transducer is installed in the stilling well. The sensor is installed 65 cm below the water level at the time of installation. The depth of water at this time is determined to be 72.6 cm relative to the desired reference. When programmed with the multiplier determined above and an offset of 0, a reading of 65.12 is obtained. The offset for the actual measurements is thus determined to be 72.6 - 65.12 = 7.48 cm.

The lead length is approximately 10 feet, so there is no appreciable error due to lead wire resistance.

```
01:      P6          Full Bridge
  01:     1          Rep
  02:    23          25 mV 60 Hz rejection Range
  03:     1          IN Chan
  04:     1          Excite all reps w/EXchan 1
  05:  2500          mV Excitation
  06:     1          Loc [:HT cm    ]
  07:    50.334      Mult
  08:     7.48       Offset
```

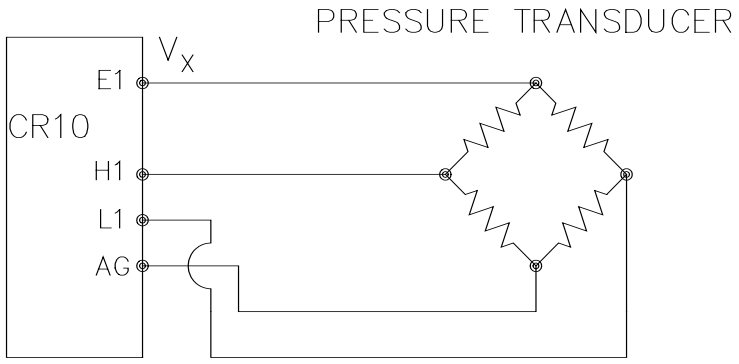


FIGURE 7.12-1. Wiring Diagram for Full Bridge Pressure Transducer

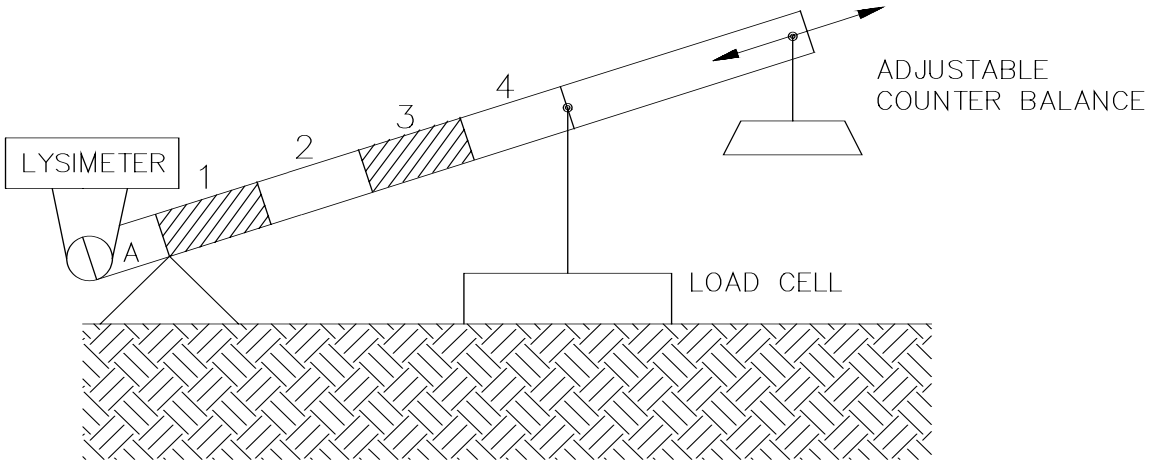


FIGURE 7.13-1. Lysimeter Weighing Mechanism

7.13 LYSIMETER - 6 WIRE FULL BRIDGE

When a long cable is required between a load cell and the CR10, the resistance of the wire can create a substantial error in the measurement if the 4 wire full bridge (Instruction 6) is used to excite and measure the load cell. This error arises because the excitation voltage is lower at the load cell than at the CR10 due to voltage drop in the cable. The 6 wire full bridge (Instruction 9) avoids this problem by measuring the excitation voltage at the load cell. This example shows the errors one would encounter if the actual excitation voltage was not measured and shows the use of a 6 wire full bridge to measure a load cell on a weighing lysimeter (a container buried in the ground, filled with plants and soil, used for measuring evapotranspiration).

The lysimeter is 2 meters in diameter and 1.5 meters deep. The total weight of the lysimeter with its container is approximately 8000 kg. The lysimeter has a mechanically adjustable counter-balance, and changes in weight are measured with a 250 pound (113.6 kg) capacity Sensotec Model 41 tension/compression load cell. The load cell has a 4:1 mechanical advantage on the lysimeter (i.e., a change of 4 kg in the mass of the lysimeter will change the force on the load cell by 1 kg-force or 980 N).

The surface area of the lysimeter is 3.1416 m^2 or 31,416 cm^2, so 1 cm of rainfall or evaporation results in a 31.416 kg change in mass. The load cell can measure ±113.6 kg, a 227 kg range. This represents a maximum change of 909 kg (28 cm of water) in the lysimeter before the counterbalance would have to be readjusted.

There is 1000 feet of 22 AWG cable between the CR10 and the load cell. The output of the load cell is directly proportional to the excitation voltage. When Instruction 6 (4 wire half bridge) is used, the assumption is that the voltage drop in the connecting cable is negligible. The average resistance of 22 AWG wire is 16.5 ohms per 1000 feet. Thus, the resistance in the excitation lead going out to the load cell added to that in the lead coming back to ground is 33 ohms. The resistance of the bridge in the load cell is 350 ohms. The voltage drop across the load cell is equal to the voltage at the CR10 multiplied by the ratio of the load cell resistance, R_S, to the total resistance, R_T, of the circuit. If Instruction 6 were used to measure the load cell, the excitation voltage actually applied to the load cell, V_1, would be:

$$V_1 = V_x \, R_s/R_T = V_x \, 350/(350+33) = 0.91 \, V_x$$

Where V_x is the excitation voltage. This means that the voltage output by the load cell would only be 91% of that expected. If recording of the lysimeter data was initiated with the load cell output at 0 volts, and 100 mm of evapotranspiration had occurred, calculation of the change with Instruction 6 would indicate that only 91 mm of water had been lost. Because the error is a fixed percentage of the output, the actual magnitude of the error increases with the force applied to the load cell. If the resistance of the wire was constant, one could correct for the voltage drop with a fixed multiplier. However, the resistance of

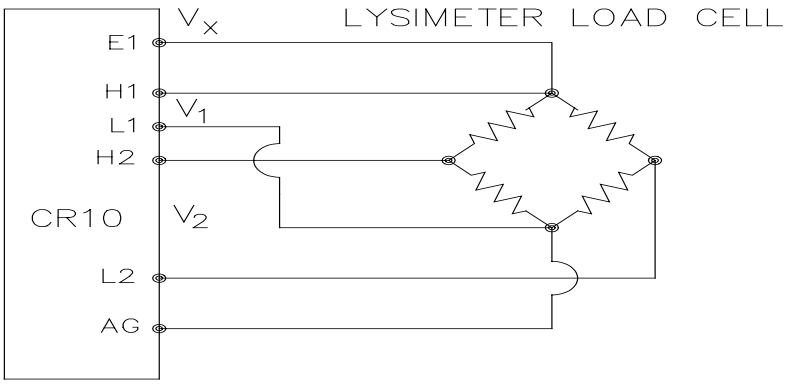


FIGURE 7.13-2. 6 Wire Full Bridge Connection for Load Cell

copper changes 0.4% per degree C change in temperature. Assume that the cable between the load cell and the CR10 lays on the soil surface and undergoes a 25°C diurnal temperature fluctuation. If the resistance is 33 ohms at the maximum temperature, then at the minimum temperature, the resistance is:

(1-25x0.004)33 ohms = 29.7 ohms

The actual excitation voltage at the load cell is:

$$V_1 = 350/(350+29.7)\ V_x = .92\ V_x$$

The excitation voltage has increased by 1%, relative to the voltage applied at the CR10. In the case where we were recording a 91 mm change in water content, there would be a 1 mm diurnal change in the recorded water content that would actually be due to the change in temperature. Instruction 9 solves this problem by actually measuring the voltage drop across the load cell bridge. The drawbacks to using Instruction 9 are that it requires an extra differential channel and the added expense of a 6 wire cable. In this case, the benefits are worth the expense.

The load cell has a nominal full scale output of 3 millivolts per volt excitation. If the excitation is 2.5 volts, the full scale output is 7.5 millivolts; thus, the ±7.5 millivolt range is selected. The calibrated output of the load cell is 3.106 mV/V_1 at a load of 250 pounds. Output is desired in millimeters of water with respect to a fixed point. The "4" found in equation 7.13-1 is due to the mechanical advantage. The calibration in mV/V_1/mm is:

$$3.106\ mV/V_1/250\ lb \times 2.2\ lb/kg \times 3.1416\ kg/mm/4 = 0.02147\ mV/V_1/mm$$

The reciprocal of this gives the multiplier to convert mV/V_1 into millimeters. (The result of Instruction 9 is the ratio of the output voltage to the actual excitation voltage multiplied by 1000, which is mV/V_1):

$$1/0.02147\ mV/V_1/mm = 46.583\ mm/mV/V_1$$

The output from the load cell is connected so that the voltage increases as the mass of the lysimeter increases. (If the actual mechanical linkage was as shown in Figure 7.13-1, the output voltage would be positive when the load cell was under tension.)

When the experiment is started, the water content of the soil in the lysimeter is approximately 25% on a volume basis. It is decided to use this as the reference (i.e., 0.25 x 1500 mm = 375 mm). The experiment is started at the beginning of what is expected to be a period during which evapotranspiration exceeds precipitation. Instruction 9 is programmed with the correct multiplier and no offset. After hooking everything up, the counterbalance is adjusted so that the load cell is near the top of its range; this will allow a longer period before readjustment is necessary. The result of Instruction 9 (monitored with the *6 Mode) is 109. The offset needed to give the desired initial value of 375 mm is 266. However, it is decided to add this offset in a separate instruction so the result of Instruction 9 can be used as a ready reminder of the strain on the load cell (range = ±140 mm). When the strain on the load cell nears its rated limits, the counterbalance is readjusted and the offset recalculated to provide a continuous record of the water budget.

The program table has an execution interval of 10 seconds. The average value in millimeters is output to Final Storage (not shown in Table) every hour.

The average is used, instead of a sample, in order to cancel out effects of wind loading on the lysimeter.

PROGRAM

```
01:      P9          Full BR w/Compensation
  01:     1          Rep
  02:    25          2500 mV 60 Hz rejection
                     EX Range
  03:    22          7.5 mV 60 Hz rejection BR
                     Range
  04:     1          IN Chan
  05:     1          Excite all reps w/EXchan 1
  06:  2500          mV Excitation
  07:     1          Loc [:RAW mm   ]
  08:  46.583        Mult
  09:     0          Offset

02:      P34         Z=X+F
  01:     1          X Loc RAW mm
  02:   266          F
  03:     2          Z Loc [:mm H20   ]
```

7.14 227 GYPSUM SOIL MOISTURE BLOCK

Soil moisture is measured with a gypsum block by relating the change in moisture to the change in resistance of the block. An AC Half Bridge (Instruction 5) is used to determine the resistance of the gypsum block. Rapid reversal of the excitation voltage inhibits polarization of the sensor. Polarization creates an error in the output so the fast integration option is used. The output of Instruction 5 is the ratio of the output voltage to the excitation voltage; this output is converted to gypsum block resistance with Instruction 59, Bridge Transform.

The Campbell Scientific 227 Soil Moisture Block uses a Delmhorst gypsum block with a 1 kohm bridge completion resistor. Using data supplied by Delmhorst, Campbell Scientific has computed coefficients for a 5th order polynomial to convert block resistance to water potential in bars. There are two polynomials: one to optimize the range from -0.1 to -2 bars, and one to cover the range from -0.1 to -10 bars (the minus sign is omitted in the output). The -0.1 to -2 bar polynomial requires a multiplier of 1 in the Bridge Transform Instruction (result in kohms) and the -0.1 to -10 bar polynomial requires a multiplier of 0.1 (result in 10,000s of ohms). The multiplier is a scaling factor to maintain the maximum number of significant digits in the polynomial coefficients.

In this example, we wish to make measurements on six gypsum blocks and output the final data in bars. The soil where the moisture measurements are to be made is quite wet at the time the data logging is initiated, but is expected to dry beyond the -2 bar limit of the wet range polynomial. The dry range polynomial is used, so a multiplier of 0.1 is entered in the bridge transform instruction.

When the water potential is computed, it is written over the resistance value. The potentials are stored in Input Locations 1-6 where they may be accessed for output to Final Storage. If it was desired to retain the resistance values, the potential measurements could be stored in Locations 7-12 by changing the value in Parameter 3 to 7 in Instruction 55. Section 8.3 gives an example using the AM416 Multiplexer to measure 16 Soil Moisture Blocks.

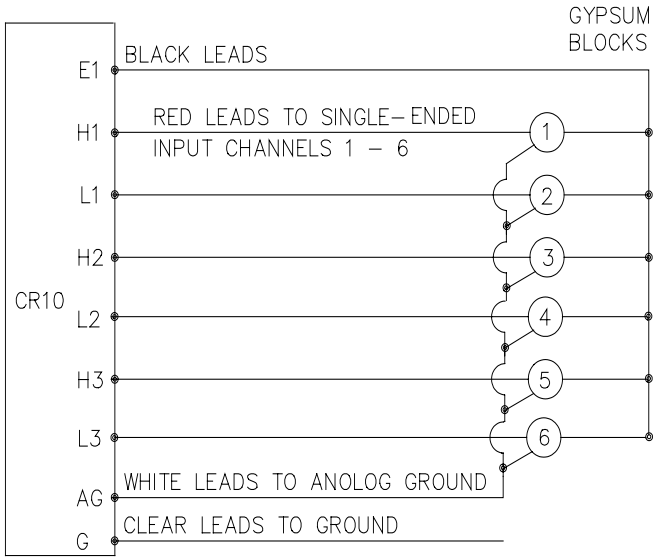


FIGURE 7.14-1. 6 227 Gypsum Blocks Connected to the CR10

PROGRAM

```
01:      P5          AC Half Bridge
  01:     6          Reps
  02:    15          2500 mV fast Range
  03:     1          IN Chan
  04:     1          Excite all reps w/EXchan 1
  05:  2500          mV Excitation
  06:     1          Loc [:H20 BARS ]
  07:     1          Mult
  08:     0          Offset

02:      P59         BR Transform Rf[X/(1-X)]
  01:     6          Reps
  02:     1          Loc [:H20 BARS ]
  03:    .1          Multiplier (Rf)

03:      P55         Polynomial
  01:     6          Reps
  02:     1          X Loc H20 BARS
  03:     1          F(X) Loc [:H20 BARS ]
  04:   .15836       C0
  05:   6.1445       C1
  06:  -8.4189       C2
  07:   9.2493       C3
  08:  -3.1685       C4
  09:   .33392       C5
```

7.15 NONLINEAR THERMISTOR IN HALF BRIDGE (MODEL 101 PROBE)

Instruction 11, 107 Thermistor Probe, automatically linearizes the output of the nonlinear thermistor in the 107 Probe by transforming the millivolt reading with a 5th order polynomial. Instruction 55, Polynomial, can be used to calculate temperature of any nonlinear thermistor, provided the correlation between temperature and probe output is known, and an appropriate polynomial fit has been determined. In this example, the CR10 is used to measure the temperature of five 101 Probes (used with the CR21 but usually not the CR10). Instruction 4, Excite, Delay, and Measure, is used because the high source resistance of the probe requires a long input settling time (Section 12.3.1). The excitation voltage is 2000 mV, the same as used in the CR21. The signal voltage is then transformed to temperature using the Polynomial Instruction.

The manual for the 101 Probe gives the coefficients of the 5th order polynomial used to convert the output in millivolts to temperature (E denotes the power of 10 by which the mantissa is multiplied):

C0	-53.7842
C1	0.147974
C2	-2.18755E-4
C3	2.19046E-7
C4	-1.11341E-10
C5	2.33651E-14

The CR10 will only allow 5 significant digits to the right or left of the decimal point to be entered from the keyboard. The polynomial cannot be applied exactly as given in the 101 manual. The initial millivolt reading must be scaled if the coefficients of the higher order terms are to be entered with the maximum number of significant digits. If 0.001 is used as a multiplier on the millivolt output, the coefficients are divided by 0.001 raised to the appropriate power (i.e., C0=C0, C1=C1/0.001, C2=C2/.000001, etc.). With this adjustment, the coefficients entered in Parameters 4-9 of Instruction 55 become:

C0	-53.784
C1	147.97
C2	-218.76
C3	219.05
C4	-111.34
C5	23.365

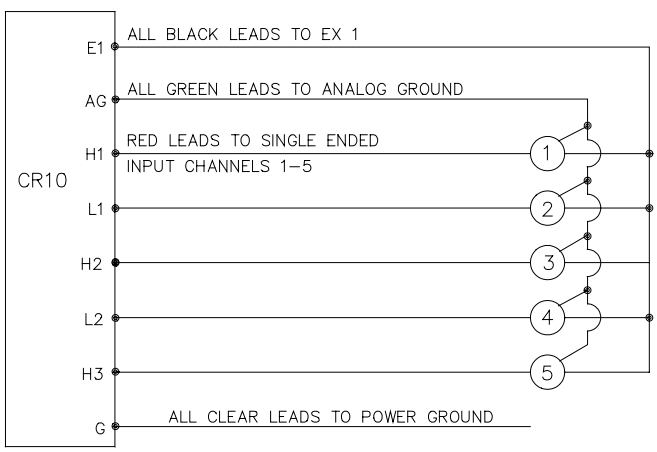


FIGURE 7.15-1. 101 Thermistor Probes Connected to CR10

PROGRAM

```
01:       P4          Excite,Delay,Volt(SE)
  01:      5          Reps
  02:     25          2500 mV 60 Hz rejection
                      Range
  03:      1          IN Chan
  04:      1          Excite all reps w/EXchan 1
  05:     10          Delay (units .01sec)
  06:   2000          mV Excitation
  07:      1          Loc [:TEMP C #1]
  08:   .001          Mult
  09:      0          Offset

02:      P55          Polynomial
  01:      5          Reps
  02:      1          X Loc TEMP C #1
  03:      1          F(X) Loc [:TEMP C #1]
  04:  -53.784        C0
  05:  147.97         C1
  06: -218.76         C2
  07:  219.05         C3
  08: -111.34         C4
  09:   23.365        C5
```

7.16 WATER LEVEL - GEOKON'S VIBRATING WIRE PRESSURE SENSOR

The vibrating wire sensor utilizes a change in the frequency of a vibrating wire to sense pressure. Figure 7.16-1 illustrates how an increase in pressure on the diaphragm decreases the tension on the wire attached to the diaphragm. A decrease in the wire tension decreases the resonant frequency in the same way that loosening a guitar string decreases its frequency.

Vibrating Wire Measurement Instruction 28 excites the "plucking" and "pickup" coils shown in Figure 7.16-1 with a "swept" frequency. A "swept" frequency is a group of different frequencies that are sent one right after another starting with the lowest frequency and ending with the highest. The lowest and highest frequencies are entered by the user in units of hundreds of Hz. This swept frequency causes the wire to vibrate at each of the individual frequencies. Ideally, all of the frequencies except the one matching the resonant frequency of the wire will die out in a very short time. The wire will vibrate with the resonant frequency for a relatively long period of time, cutting the lines of flux in the "plucking" and "pickup" coils and inducing the same frequency on the lines to the CR10. Instruction 28 then accurately measures how much time it takes to receive a user specified number of cycles.

The vibrating wire requires temperature compensation. A nonlinear thermistor built into the probe is measured using Instruction 4, a single-ended half bridge measurement with excitation, and calculated with Instruction 55, a fifth order polynomial instruction.

Campbell Scientific's AVW1 or AVW4 Vibrating Wire Sensor Interface is required between the sensor to the datalogger. The purpose is twofold:

- 12 volts can be used as the potential in the swept frequency excitation, thus plucking the wire harder than the maximum 2.5 volt switched excitation. The result is a larger magnitude signal for a longer time.
- A transformer strips off any DC noise on the signal, improving the ability to detect cycles.

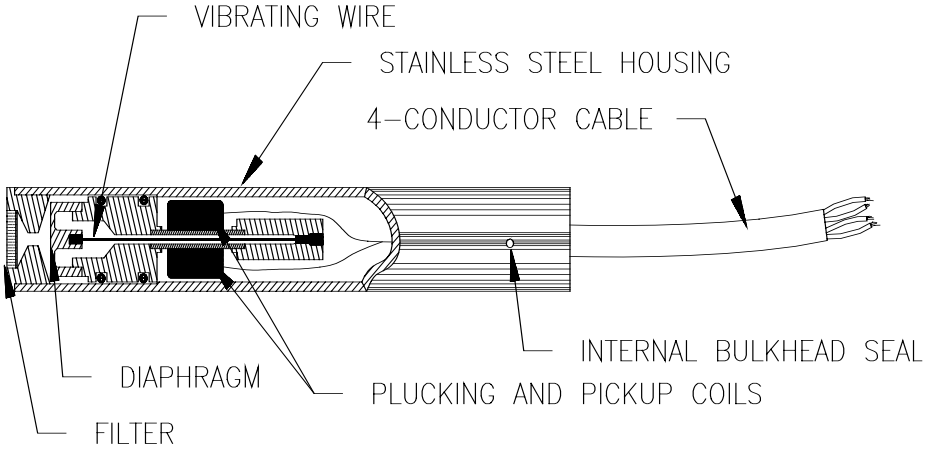


FIGURE 7.16-1. A Vibrating Wire Sensor

The following calculations are based on using a Geokon model 4500 Vibrating Wire sensor. An individual multiplier and offset must be calculated for each sensor used in a system.

MULTIPLIER

The fundamental equation relating frequency to pressure is

$$P = -F_x G + B \quad \text{where}$$

P = pressure, PSI
G = the Gage Factor obtained from the sensors calibration sheet in PSI/digit. The units of a digit are $Hz^2(10^{-3})$.
B = offset
$F_x = f^2Hz^2(10^{-3})$, where f is frequency.

Instruction 28 measures period, T, of the vibrating wire in milliseconds (ms) and returns a measured value, X, of

$$X = 1/(T^2(ms)^2) = f^2(10^{-6})Hz^2$$

A multiplier of -1000 in Instruction 28 converts the measurement to digits, as shown below.

$$-F_x = -X(-10^3) = -f^2(10^{-3})Hz^2$$

To calculate the multiplier, convert Geokon's gage factor, G, to the desired units (i.e., feet of water per digit) and multiply by -1000.

TEMPERATURE CORRECTION

The temperature correction is applied as follows.

$$P_T = P + C * (t_1 - t_0), \text{ where}$$

P_T = Pressure corrected for temperature, °C
C = Temperature coefficient, PSI/°C (from Geokon calibration sheet)
t_0 & t_1 = Initial and current temperatures, °C.

The temperature coefficient, C, must be converted to units compatible with the gage factor, G.

WELL MONITORING EXAMPLE

In this example the vibrating wire sensor is used to monitor water table height (Figure 7.16-2). The desired data is the distance from the lip of the well to the water surface. The sensor is vented to atmosphere to eliminate measurement errors due to changes in barometric pressure. The water level is expected to stay within 40 to 80 feet of the lip so the 50 psi pressure sensor is placed approximately 100 feet below the lip of the well. The calibration data from Geokon is provided in Table 7.16-1.

TABLE 7.16-1 Calibration Data for Sensor 3998

Gage Factor (psi/digit)	Temp. Coeff. (psi/°C)
0.0151	-0.0698

The multiplier, m, is calculated to convert the reading to feet of water.

m = 0.0151 (psi/digit) * 2.3067 (ft of water/psi) * -1000 = -34.831 ft of water/digit

After the probe reaches thermal equilibrium, the initial temperature, t_0, is measured to be 24°C.

The water column above the sensor is referred to as the "Reading". The Reading decreases with increasing "Distance" from lip of well to water surface so the Distance is computed by subtracting the Reading from the Offset as shown in Figure 7.16-2.

The "Initial Distance" to the water surface is measured with a chalked line to be 47.23 feet below the lip. The first time the program is executed, the program calculates the offset (Offset = Distance + Reading) required to obtain a reading of 47.23 feet. The offset is stored in Location 4 and applied to subsequent measurements.

NOTE: Following program compilation in the *0 Mode, all input locations are set to zero. This fact is utilized to detect the first execution following a program compilation.

The example assumes the sensor has been connected as shown here.

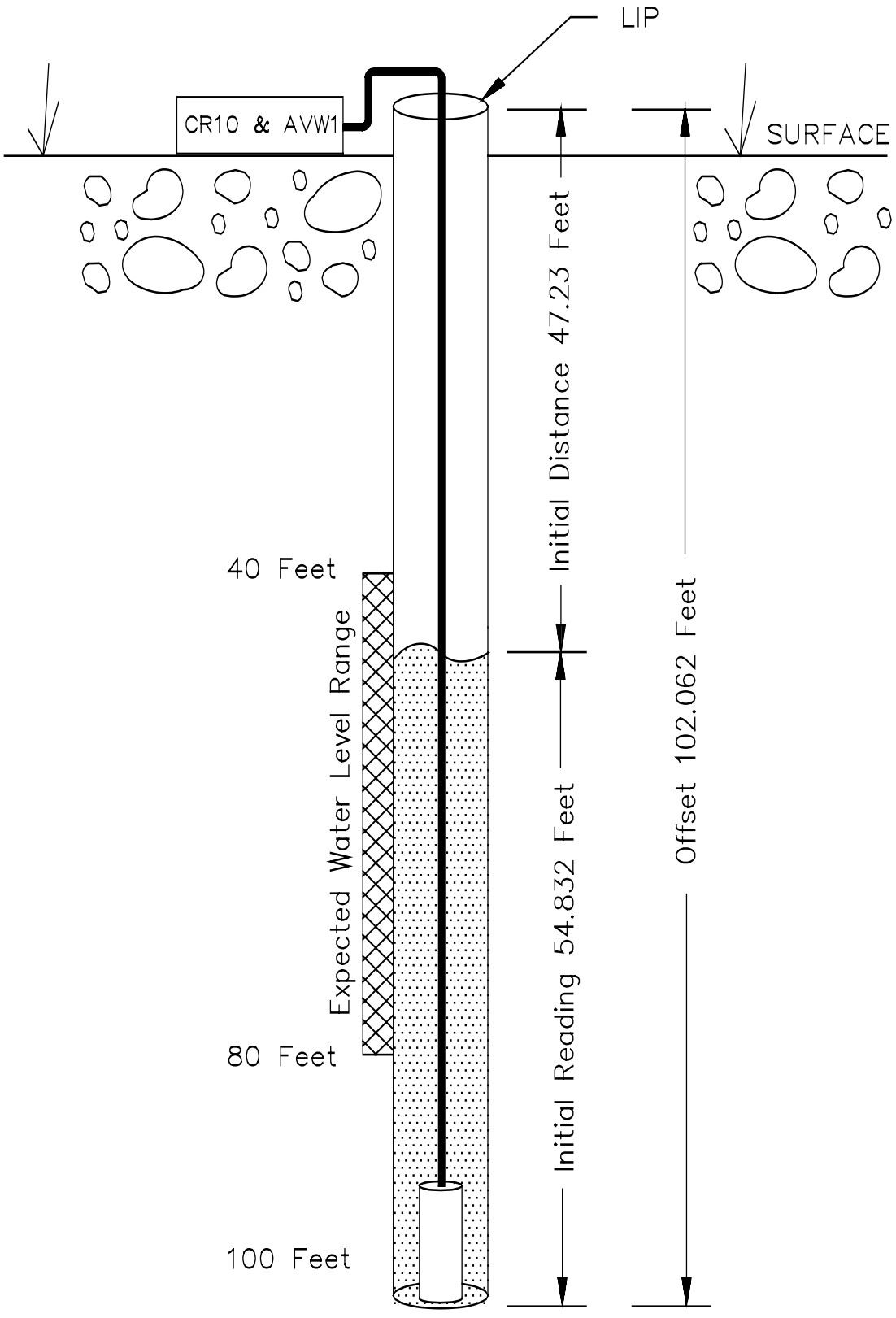


FIGURE 7.16-2. Well Monitoring Example

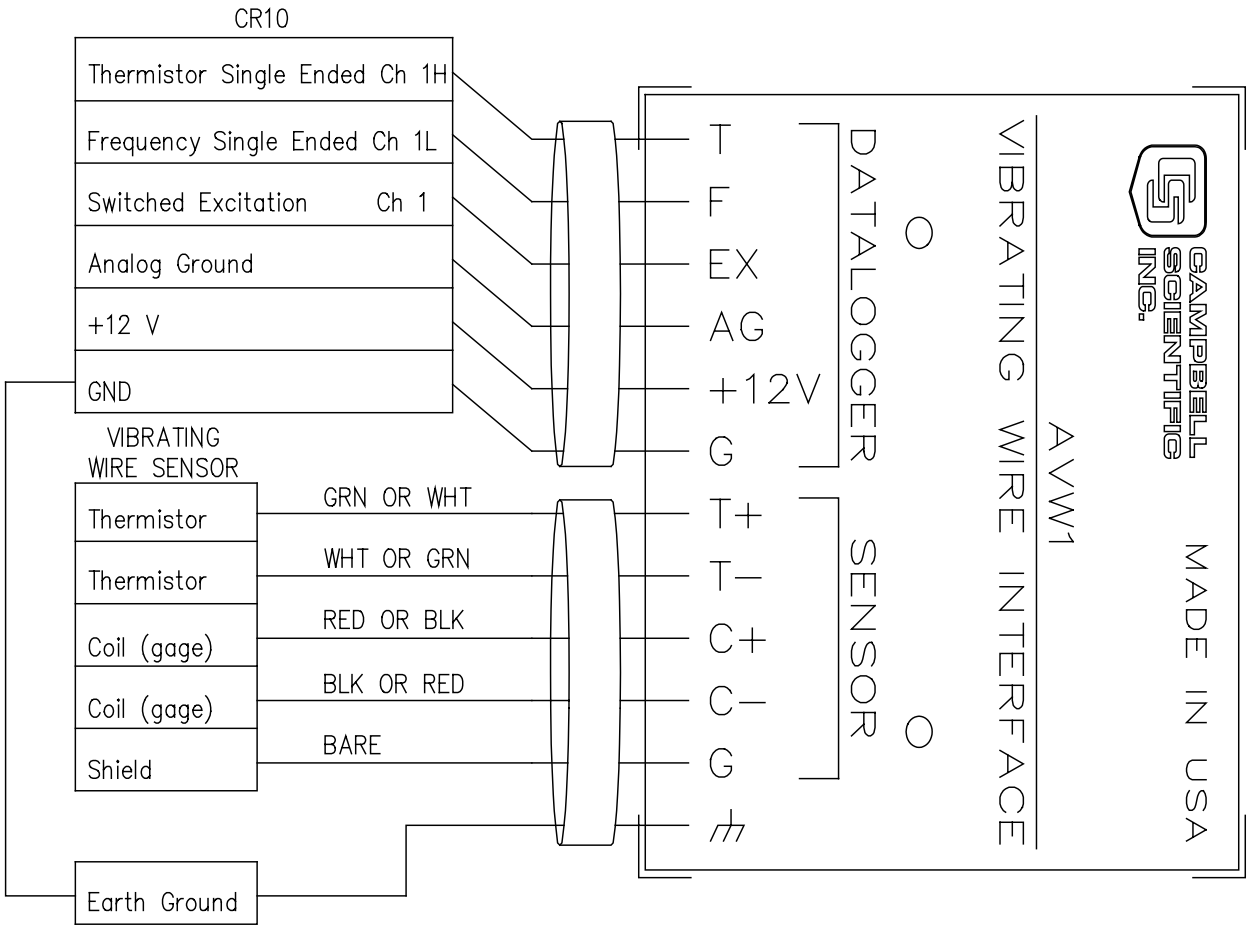


FIGURE 7.16-3. Hook up to AVW1

Program: AVW1 & CR10 USED TO MEASURE 1 GEOKON VIBRATING WIRE SENSOR.

*	1	Table 1 Programs
01:	60	Sec. Execution Interval
01:	P4	Excite,Delay,Volt(SE)
01:	1	Rep
02:	15	2500 mV fast Range
03:	1	IN Chan
04:	1	Excite all reps w/EXchan 1
05:	1	Delay (units .01sec)
06:	2500	mV Excitation
07:	1	Loc [:TEMP]
08:	.001	Mult
09:	0	Offset
02:	P55	Polynomial
01:	1	Rep
02:	1	X Loc TEMP
03:	1	F(X) Loc [:TEMP]
04:	-104.78	C0
05:	378.11	C1
06:	-611.59	C2
07:	544.27	C3
08:	-240.91	C4
09:	43.089	C5
03:	P28	Vibrating Wire (SE)
01:	1	Rep
02:	2	IN Chan
03:	1	Excite all reps w/EXchan 1
04:	24	Starting Freq. (units=100 Hz)
05:	32	End Freq. (units=100 Hz)
06:	500	No. of Cycles
07:	0	Rep delay (units=.01sec)
08:	2	Loc [:PRESSURE]
09:	-34.836	Mult

```
 10:      0          Offset

04:      P34        Z=X+F
 01:       1         X Loc TEMP
 02:     -24         F
 03:       3         Z Loc [:TEMP COMP]

05:      P37        Z=X*F
 01:       3         X Loc TEMP COMP
 02:     -.0698      F
 03:       3         Z Loc [:TEMP COMP]

06:      P33        Z=X+Y
 01:       3         X Loc TEMP COMP
 02:       2         Y Loc PRESSURE
 03:       2         Z Loc [:PRESSURE ]

07:      P89        If X<=>F
 01:       5         X Loc CMPILE CK
 02:       1         =
 03:       0         F
 04:      30         Then Do

08:      P34        Z=X+F
 01:       2         X Loc PRESSURE
 02:   47.23         F
 03:       4         Z Loc [:OFFSET   ]

09:      P30        Z=F
 01:       1         F
 02:       0         Exponent of 10
 03:       5         Z Loc [:CMPILE CK]

10:      P95        End

11:      P35        Z=X-Y
 01:       4         X Loc OFFSET
 02:       2         Y Loc PRESSURE
 03:       6         Z Loc [:DISTANCE ]

12:        P        End Table 1
```

7.17 PAROSCIENTIFIC "T" SERIES PRESSURE TRANSDUCER

The Paroscientific "T" series transducer has two resonating quartz crystals that output frequency signals for temperature and pressure. The pressure output requires temperature compensation. Instruction 27 measures the outputs and returns period in microseconds (refer to Table 7.17-1).

TABLE 7.17-1 Period Averaging Inst. 27

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Gain/output option
03:	2	Single Ended Input Channel
04:	4	# Cycles to measure
05:	4	Time out (0.01 sec, at least the maximum duration of the # of cycles specified + 1 1/2 cycles.)
06:	4	Destination input location
07:	FP	Multiplier
08:	FP	Offset

GAIN/OUTPUT OPTION, PARAMETER 2

The nominal pressure frequency range is 32 to 42 kHz over the stated pressure range (period of $3.1*10^{-5}$ to $2.4*10^{-5}$). The nominal temperature frequency is 172 kHz, ±50 ppm/°C (period of $5.8*10^{-6}$). The gain/output option (Parameter 2) is based on the maximum input frequency. For the stated temperature and pressure frequency ranges, the gain codes are 3 and 4 (refer to Table 7.17-2). Codes 1 and 2 are not used with Paroscientific transducers. For period, the output option is 0.

TABLE 7.17-2 Input Frequency Gain Codes

Range Code	Peak to Peak Volts Required @ Max. Freq.*	Maximum Frequency
1	2 mV	8 kHz
2	3 mV	20 kHz
3	12 mV	50 kHz
4	2 V	200 kHz

0x Output period in microseconds
1x Output frequency in kHz
where x is range code

***** AC voltage; must be centered around CR10 ground.

TIME OUT, PARAMETER 5

The "time out", Parameter 5, specifies the maximum length of time the instruction waits on each repetition to receive the number of cycles specified in Parameter 4. The time out units are 0.01 seconds. The minimum time out is the time required to receive the specified number of cycles at the maximum expected frequency.

Time out calculations using a recommended 9000 and 5000 cycles for temperature and pressure at the maximum frequency are shown below.

Time out for temperature: 6,
5.22 = $(5.8*10^{-6})(9000/0.01)$

Time out for pressure: 16,
15.5 = $(3.1*10^{-5})(5000/0.01)$

If the time out expires before the requested number of cycles are read, -99999 is stored in the input location (Parameter 6). After reading the correct number of cycles, program execution advances to the next instruction; the time out may be over estimated without losing execution speed.

CONVERTING MICROSECONDS TO °C AND PSI

The temperature and pressure in microseconds are converted to °C and psi according to the following relationships provided by Paroscientific.

Pressure (psi) =
$C * (1 \ T_0^2/Tau^2) * (1 \ (D * (1 - T_0^2/Tau^2)))$

Temperature (°C) = $Y_1U + Y_2U^2 + Y_3U^3$

where, $C = C_1 + C_2U + C_3U^2$ (psi),

$T_0 = T_1 + T_2U + T_3U^2 + T_4U^3 + T_5U^4$ (microsecond),

$D = D_1 + D_2U$ (microsecond),

$U = U(t) \ U_0$ (microsecond),

Tau = measured pressure (microsecond),

U(t) = measured temperature (microsecond).

Values for the calibration coefficients (U_0, Y_1, Y_2, Y_3, C_1, C_2, C_3, D_1, D_2, T_1, T_2, T_3, T_4, T_5) are provided by Paroscientific.

ENTERING THE COEFFICIENTS

Coefficients are entered using CR10 Instruction 30 or 64. A calibration sheet from Paroscientific which accompanies the transducer lists the 14 coefficients. Table 7.17-3 shows the correct format for entering the coefficients. The coefficients shown are for transducer Serial Number 30135. Your coefficients will be different.

TABLE 7.17-3 Coefficient Entry Format for Paroscientific "T" Series Pressure Transducer Instruction 30

Coeff.	Value	Param. 1	Param. 2
U_0	5.860253	5.8603	0.0
Y_1	-3970.348	-3970.3	0.0
Y_2	-7114.265	-7114.3	0.0
Y_3	102779.1	102.78	3.0*
C_1	70.29398	70.294	0.0
C_2	6.610141	6.6101	0.0
C_3	-119.2867	-119.29	0.0
D_1	0.0308837	30.884	-3.0*
D_2	0.0	0.0	0.0
T_1	26.33703	26.337	0.0
T_2	0.8516985	0.85170	0.0
T_3	21.80118	21.801	0.0
T_4	0.0	0.0	0.0
T_5	0.0	0.0	0.0

**Y_3 and D_1 are the only coefficients with a non-zero exponent.*

Only the first 5 digits of each 7 digit coefficient are entered in the datalogger. The maximum error that occurs due to using five digit coefficients is 0.001 psi throughout a 15 psi range at any temperature between -30 and +60°C. The maximum error in temperature calculations is 0.2°C over the same temperature range.

CONNECTIONS

Figure 7.17-1 details the components required for connecting the transducer to the CR10. The user supplied components are commonly available at commercial electronic stores.

PROGRAM EXAMPLE

The program may be divided into the following 7 steps.

1. Load the coefficients into input locations (Subroutine 1).
2. Measure the temperature and pressure.
3. Compute U.
4. Compute temperature, °C.
5. Compute T_0, T_0^2, Tau^2, and $1-T_0^2/Tau^2$.
6. Compute D.
7. Compute C.
8. Compute pressure (psi.)

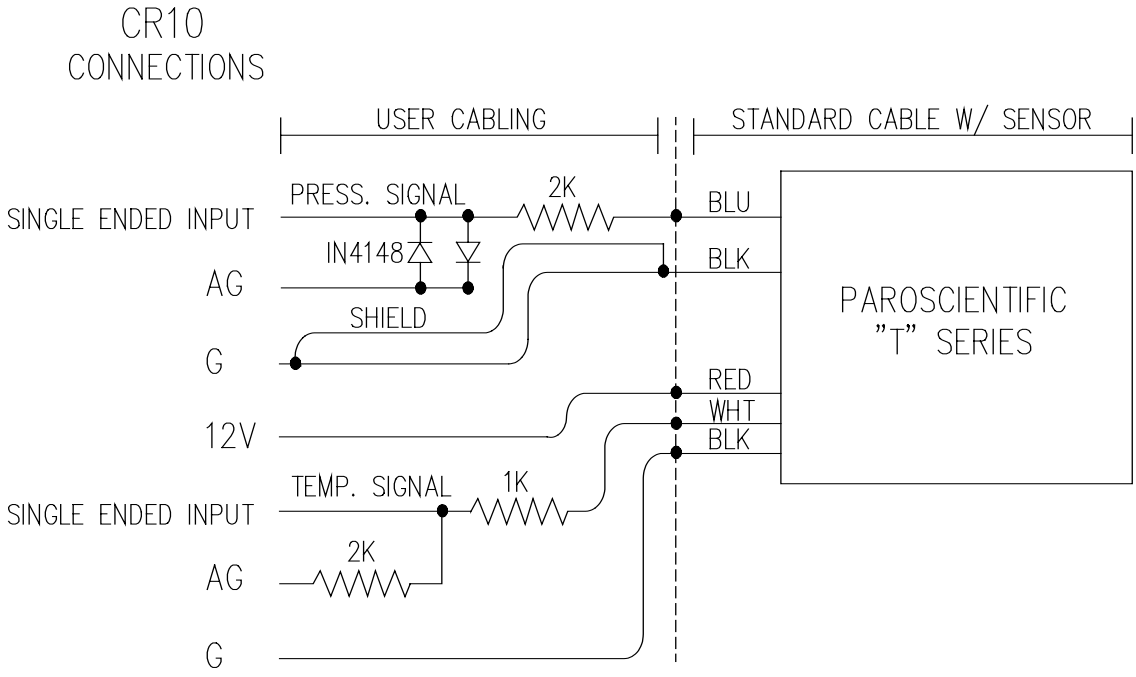


FIGURE 7.17-1. CR10/Paroscientific "T" Series Transducer Wiring Diagram

Subroutine 1, which loads the coefficients into input locations, is called only on the first execution following program compilation.

The temperature frequency is read on single-ended Channel 12 and pressure is measured on single-ended Channel 1.

Temperature, T_0, D and C are computed in Subroutine 2 using a generalized fourth order polynomial equation. The form of the equation is

$F(x) = U_0 + U(M_1 + U(M_2 + U(M_3 + UM_4)))$.

This form provides faster execution and greater accuracy of the floating point math. "$M_{0..4}$" represents coefficients 0 through 4, and U is defined above. With five coefficients, T_0 is the only parameter requiring a fourth order polynomial. To maintain the fourth order format, zero value "dummy" coefficients are plugged in where required for the temperature, D, and C coefficients.

Instructions to output the readings to Final Storage are not included in this example.

```
*          1     Table 1 Programs
  01:     60     Sec. Execution Interval

01:       P89    If X<=>F
  01:     42     X Loc CMPILE CK
  02:      1     =
  03:      0     F
  04:      1     Call Subroutine 1

02:       P27    Period Average (SE)
  01:      1     Rep
  02:      4     Input gain=1
  03:     12     IN Chan
  04:   9000     No. of Cycles
  05:      6     Time Out (units=.01sec)
  06:     10     Loc [:UT us    ]
  07:      1     Mult
  08:      0     Offset

03:       P27    Period Average (SE)
  01:      1     Rep
  02:      3     Input gain=10
  03:      1     IN Chan
  04:   5000     No. of Cycles
  05:     16     Time Out (units=.01sec)
  06:     11     Loc [:TAU us   ]
  07:      1     Mult
  08:      0     Offset
```

```
04:   P35     Z=X-Y
 01:   10     X Loc UT us
 02:    9     Y Loc Uo
 03:    8     Z Loc [:U       ]

05:   P54     Block Move
 01:    5     No. of Values
 02:   20     First Source Loc Y4
              DUMMY
 03:    1     Source Step
 04:   15     Destination Loc [:POLLY
              M4 ]
 05:    1     Destination Step

06:   P86     Do
 01:    2     Call Subroutine 2

07:   P31     Z=X
 01:   40     X Loc SCRATCH 1
 02:    1     Z Loc [:TEMP C   ]

08:   P54     Block Move
 01:    5     No. of Values
 02:   35     First Source Loc T5
 03:    1     Source Step
 04:   15     Destination Loc [:POLLY
              M4 ]
 05:    1     Destination Step

09:   P86     Do
 01:    2     Call Subroutine 2

10:   P31     Z=X
 01:   40     X Loc SCRATCH 1
 02:   12     Z Loc [:TEMP C   ]

11:   P36     Z=X*Y
 01:   11     X Loc TAU us
 02:   11     Y Loc TAU us
 03:    6     Z Loc [:TAU ^2   ]

12:   P36     Z=X*Y
 01:   12     X Loc To us
 02:   12     Y Loc To us
 03:    7     Z Loc [:To^2     ]

13:   P38     Z=X/Y
 01:    7     X Loc To^2
 02:    6     Y Loc TAU ^2
 03:   40     Z Loc [:SCRATCH 1]

14:   P30     Z=F
 01:    1     F
 02:    0     Exponent of 10
 03:   41     Z Loc [:SCRATCH 2]

15:   P35     Z=X-Y
 01:   41     X Loc SCRATCH 2
 02:   40     Y Loc SCRATCH 1
 03:    5     Z Loc [:1-(T/Ta)^]

16:   P54     Block Move
 01:    5     No. of Values
 02:   30     First Source Loc D5
              DUMMY
 03:    1     Source Step
 04:   15     Destination Loc [:POLLY
              M4 ]
 05:    1     Destination Step

17:   P86     Do
 01:    2     Call Subroutine 2

18:   P31     Z=X
 01:   40     X Loc SCRATCH 1
 02:   13     Z Loc [:D        ]

19:   P54     Block Move
 01:    5     No. of Values
 02:   25     First Source Loc C5
              DUMMY
 03:    1     Source Step
 04:   15     Destination Loc [:POLLY
               M4 ]
 05:    1     Destination Step

20:   P86     Do
 01:    2     Call Subroutine 2

21:   P31     Z=X
 01:   40     X Loc SCRATCH 1
 02:   14     Z Loc [:C        ]

22:   P36     Z=X*Y
 01:    5     X Loc 1-(T/Ta)^
 02:   13     Y Loc D
 03:   40     Z Loc [:SCRATCH 1]

23:   P35     Z=X-Y
 01:   41     X Loc SCRATCH 2
 02:   40     Y Loc SCRATCH 1
 03:   40     Z Loc [:SCRATCH 1]

24:   P36     Z=X*Y
 01:    5     X Loc 1-(T/Ta)^
 02:   40     Y Loc SCRATCH 1
 03:   40     Z Loc [:SCRATCH 1]

25:   P36     Z=X*Y
 01:   14     X Loc C
 02:   40     Y Loc SCRATCH 1
 03:    2     Z Loc [:PRESS PSI]
```

```
26:          P          End Table 1

*            3          Table 3 Subroutines

01:        P85          Beginning of Subroutine
  01:        1          Subroutine Number

02:        P30          Z=F
  01:     5.8603          F
  02:        0          Exponent of 10
  03:        9          Z Loc [:Uo      ]

03:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       24          Z Loc [:Y0 DUMMY ]

04:        P30          Z=F
  01:  -3970.3          F
  02:        0          Exponent of 10
  03:       23          Z Loc [:Y1      ]

05:        P30          Z=F
  01:  -7114.3          F
  02:        0          Exponent of 10
  03:       22          Z Loc [:Y2      ]

06:        P30          Z=F
  01:   102.78          F
  02:        3          Exponent of 10
  03:       21          Z Loc [:Y3      ]

07:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       20          Z Loc [:Y4 DUMMY ]

08:        P30          Z=F
  01:    70.294          F
  02:        0          Exponent of 10
  03:       29          Z Loc [:C1      ]

09:        P30          Z=F
  01:    6.6101          F
  02:        0          Exponent of 10
  03:       28          Z Loc [:C2      ]

10:        P30          Z=F
  01:  -119.29          F
  02:        0          Exponent of 10
  03:       27          Z Loc [:C3      ]

11:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       26          Z Loc [:C4 DUMMY ]

12:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       25          Z Loc [:C5 DUMMY ]

13:        P30          Z=F
  01:   30.884          F
  02:        3--        Exponent of 10
  03:       34          Z Loc [:D1      ]

14:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       33          Z Loc [:D2      ]

15:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       32          Z Loc [:D3 DUMMY ]

16:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       31          Z Loc [:D4 DUMMY ]

17:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       30          Z Loc [:D5 DUMMY ]

18:        P30          Z=F
  01:   26.337          F
  02:        0          Exponent of 10
  03:       39          Z Loc [:T1      ]

19:        P30          Z=F
  01:      .8517          F
  02:        0          Exponent of 10
  03:       38          Z Loc [:T2      ]

20:        P30          Z=F
  01:   21.801          F
  02:        0          Exponent of 10
  03:       37          Z Loc [:T3      ]

21:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       36          Z Loc [:T4      ]

22:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       35          Z Loc [:T5      ]
```

```
23:     P30         Z=F
  01:     0         F
  02:     0         Exponent of 10
  03:    40         Z Loc [:SCRATCH 1]

24:     P30         Z=F
  01:     0         F
  02:     0         Exponent of 10
  03:    41         Z Loc [:SCRATCH 2]

25:     P30         Z=F
  01:     0         F
  02:     0         Exponent of 10
  03:    42         Z Loc [:CMPILE CK]

26:     P95         End

27:     P85         Beginning of Subroutine
  01:     2         Subroutine Number
28:       P         36  Z=X*Y
  01:    15         X Loc POLLY M4
  02:     8         Y Loc U
  03:    40         Z Loc [:SCRATCH 1]

29:     P33         Z=X+Y
  01:    16         X Loc POLY M3
  02:    40         Y Loc SCRATCH 1
  03:    40         Z Loc [:SCRATCH 1]

30:     P87         Beginning of Loop
  01:     0         Delay
  02:     3         Loop Count

31:     P36         Z=X*Y
  01:     8         X Loc U
  02:    40         Y Loc SCRATCH 1
  03:    40         Z Loc [:SCRATCH 1]

32:     P33         Z=X+Y
  01:   17--        X Loc POLY M2
  02:    40         Y Loc SCRATCH 1
  03:    40         Z Loc [:SCRATCH 1]

33:     P95         End

34:     P95         End

35:       P         End Table 3


*         A         Mode 10 Memory
                    Allocation
  01:    50         Input locations
  02:    64         Intermediate locations
  03:  0.0000       Final Storage Area 2
```

7.18 SDM PERIPHERALS

The SDM peripherals are measurement and control modules which are controlled by the CR10 through control ports 1, 2, and 3. The instructions for these peripherals are: 101 SDM-INT8 8 channel interval timer, 102 SDM-SW8 8 channel switch closure multiplexer, 103 SDM-A04 4 channel analog output multiplexer, and 104 SDM-CD16 16 channel control port expansion. Please consult the SDM peripheral manual for programming examples.

7.19 PAROSCIENTIFIC PRESSURE TRANSDUCER PROCESSING

This example demonstrates the use of Instruction 64 and 65 for calculating the pressure measured with a Paroscientific "T" Series transducer. Figure 7.17-1 details the components required for connecting the transducer to the CR10. The user-supplied components are commonly available at commercial electronic stores.

Example: The 14 coefficients shown below are for Paroscientific "T" Series transducer Serial Number 30135. Your coefficients will be different.

Coeff.	Value	Entry
U_0	5.860253	5.8603
Y_1	-3970.348	-3970.3
Y_2	-7114.265	-7114.3
* Y_3	102779.1	102.78
C_1	70.29398	70.294
C_2	6.610141	6.6101
C_3	-119.2867	-119.29
* D_1	0.0308837	30.884
D_2	0.0	0.0
T_1	26.33703	26.337
T_2	0.8516985	0.85170
T_3	21.80118	21.801
T_4	0.0	0.0
T_5	0.0	0.0

* Y_3 and D_1 coefficients are entered as $Y_3/1000$ and D_1*1000.

The following example reads the coefficients from a subroutine only when the datalogger program is compiled. The coefficients are stored in Input Locations 3 through 16. The temperature frequency is read on single-ended Channel 12 and stored in Input Location 1. Pressure is measured on single-ended Channel 1 and stored in Location 2. Instruction 64

converts the readings to engineering units. Temperature (°C), pressure (psi), and signature are stored in Locations 17, 18, and 19, respectively. Instructions to output the readings to Final Storage are not included in this example.

```
*            1          Table 1 Programs
  01:       60          Sec. Execution Interval
```

If the program has just compiled, a 0 is in Loc 20. If Loc 20 = 0, call subroutine 1 and load the temperature coefficients into Loc 3..16.

```
01:        P89          If X<=>F
  01:       20          X Loc COMP CHK
  02:        1          =
  03:        0          F
  04:        1          Call Subroutine 1
```

Measure temperature period in microseconds

```
02:        P27          Period Average (SE)
  01:        1          Rep
  02:        4          Input gain=1
  03:       12          IN Chan
  04:     9000          No. of Cycles
  05:        6          Time Out (units=.01sec)
  06:        1          Loc [:TEMP,us]
  07:        1          Mult
  08:        1          Offset
```

Measure pressure period in microseconds

```
03:        P27          Period Average (SE)
  01:        1          Rep
  02:        4          Input gain=1
  03:        1          IN Chan
  04:     5000          No. of Cycles
  05:       15          Time Out (units=.01sec)
  06:        2          Loc [:PRESS,us]
  07:        1          Mult
  08:        0          Offset

04:        P64          Paro Processing
01:          1          Start Loc: Temp(us),
                        Pressure(us), U0..T5
02:         17          Start Loc: Temp(°C),
                        Pressure(psi), SIG

05:          P          End Table 1


*            3          Table 3 Subroutines

01:        P85          Beginning of Subroutine
  01:        1          Subroutine Number

02:        P30          Z=F
  01:     5.8603        F
  02:        0          Exponent of 10
  03:        3          Z Loc [:U0   ]

03:        P30          Z=F
  01:  -3970.3          F
  02:        0          Exponent of 10
  03:        4          Z Loc [:Y1   ]

04:        P30          Z=F
  01:  -7114.3          F
  02:        0          Exponent of 10
  03:        5          Z Loc [:Y2   ]

05:        P30          Z=F
  01:   102.78          F
  02:        0          Exponent of 10
  03:        6          Z Loc [:Y3   ]

06:        P30          Z=F
  01:    70.294         F
  02:        0          Exponent of 10
  03:        7          Z Loc [:C1   ]

07:        P30          Z=F
  01:    6.6101         F
  02:        0          Exponent of 10
  03:        8          Z Loc [:C2   ]

08:        P30          Z=F
  01:  -119.29          F
  02:        0          Exponent of 10
  03:        9          Z Loc [:C3   ]

09:        P30          Z=F
  01:    30.884         F
  02:        0          Exponent of 10
  03:       10          Z Loc [:D1   ]

10:        P30          Z=F
  01:        0          F
  02:        0          Exponent of 10
  03:       11          Z Loc [:D2   ]

11:        P30          Z=F
  01:    26.337         F
  02:        0          Exponent of 10
  03:       12          Z Loc [:T1   ]

12:        P30          Z=F
  01:      .85170        F
  02:        0          Exponent of 10
  03:       13          Z Loc [:T2   ]
```

```
13:     P30         Z=F
  01:     21.801    F
  02:      0        Exponent of 10
  03:     14        Z Loc [:T3   ]

14:     P30         Z=F
  01:      0        F
  02:      0        Exponent of 10
  03:     15        Z Loc [:T4   ]

15:     P30         Z=F
  01:      0        F
  02:      0        Exponent of 10
  03:     16        Z Loc [:T5   ]

16:     P30         Z=F
  01:      1        F
  02:      0        Exponent of 10
  03:     20        Z Loc [:COMP CHK]

17:     P95         End

18:       P         End Table 3
```

INPUT LOCATION LABELS:

1:TEMP, us	11:D2
2:PRESS us	12:T1
3:U0	13:T2
4:Y1	14:T3
5:Y2	15:T4
6:Y3	16:T5
7:C1	17:TEMP, (°C)
8:C2	18:psi
9:C3	19:SIG
10:D1	20:COMP CHK

SECTION 8. PROCESSING AND PROGRAM CONTROL EXAMPLES

The following examples are intended to illustrate the use of Processing and Program Control Instructions, flags, dual Final Storage, and the capability to direct the results of Output Processing Instructions to Input Storage.

The specific examples may not be as important as some of the techniques employed, for example:

Directing Output Processing to Input Storage is used in the Running Average and Rainfall Intensity examples (8.1 and 8.2).

Flag tests are used in the Running Average, Interrupt Subroutine, Converting Wind Direction, and Saving Data Prior to Event examples (8.1, 8.5, 8.7 and 8.8).

Control ports and the Loop are illustrated in the AM32 example (8.3).

An algorithm for a down counter is used in the Saving Data Prior to Event example (8.8).

As in Section 7 these examples are not complete programs to be taken verbatim. They need to be altered to fit specific needs.

8.1 COMPUTATION OF RUNNING AVERAGE

It is sometimes necessary to compute a running average (i.e., the average covers a fixed number of samples and is continuously updated as new samples are taken). Because the output interval is shorter than the averaging period, Instruction 71 cannot be used; the algorithm for computing this average must be programmed by the user. The following example demonstrates a program for computing a running average.

In this example, each time a new measurement is made (in this case a thermocouple temperature) an average is computed for the 10 most recent samples. This is done by saving all 10 temperatures in contiguous input locations and using the Spatial Average Instruction (51) to compute the average. The temperatures are stored in locations 11 through 20. Each time the table is executed, the new measurement is stored in location 20 and the average is stored in location 2. The Block Move Instruction (54) is then used to move the temperatures from locations 12 through 20 down by 1 location; the oldest measurement (in location 11) is lost when the temperature from location 12 is written over it.

Input Location Labels:

1:Panl Temp	15:Temp i-5
2:10smpl av	16:Temp i-4
11:Temp i-9	17:Temp i-3
12:Temp i-8	18:Temp i-2
13:Temp i-7	19:Temp i-1
14:Temp i-6	20:Temp i

Where i is current reading,
i-1 is previous reading, etc.

```
*          1      Table 1 Programs
  01:     10      Sec. Execution Interval

01:       P17     Panel Temperature
  01:      1      Loc [:Panl Temp]

02:       P14     Thermocouple Temp (DIFF)
  01:      1      Rep
  02:      1      2.5 mV slow Range
  03:      1      IN Chan
  04:      1      Type T (Copper-Constantan)
  05:      1      Ref Temp Loc Panl Temp
  06:     20      Loc [:Temp i   ]
  07:      1      Mult
  08:      0      Offset

03:       P51     Spatial Average
  01:     10      Swath
  02:     11      First Loc Temp i-9
  03:      2      Avg Loc [:10smpl av]
```

```
04:     P54     Block Move
  01:     9     No. of Values
  02:    12     First Source Loc Temp i-8
  03:     1     Source Step
  04:    11     First Dest. Loc [:Temp i-9 ]
  05:     1     Destination Step

05:     P86     Do
  01:    10     Set high Flag 0 (output)

06:     P70     Sample
  01:     1     Reps
  02:     2     Loc 10smpl av

07:       P     End Table 1
```

In the above example, all samples for the average are stored in input locations. This is necessary when an average must be output with each new sample. In most cases, averages are desired less frequently than sampling. For example, it may be necessary to sample some parameter every 5 seconds and output every hour an average of the previous three hours' readings. If all samples were saved, this would require 2160 input locations. The same value can be obtained by computing an hourly average and averaging the hourly averages for the past three hours. To do this requires that hourly averages be stored in input locations.

Instruction 80 is used to send the 1 hour average to Input Storage and again to send the 3 hour average to Final Storage.

Input Location Labels:

```
1:AVG i-2
2:AVG i-1
3:AVG i
4:3 HR AVG
5:XX mg/M3
```

```
*          1     Table 1 Programs
  01:      5     Sec. Execution Interval

01:       P2     Volt (DIFF)
  01:      1     Rep
  02:     25     2500 mV 60 Hz rejection
  03:      3     IN Chan
  04:      5     Loc [:XX mg/M3 ]
  05:     10     Mult
  06:      0     Offset

02:     P92     If time is
  01:      0     minutes into a
  02:     60     minute interval
  03:     10     Set high Flag 0 (output)

03:     P80     Set Active Storage Area
  01:      3     Input Storage Area
  02:      3     Array ID or location

04:     P71     Average
  01:      1     Rep
  02:      5     Loc XX mg/M3

05:     P51     Spatial Average
  01:      3     Swath
  02:      1     First Loc AVG i-2
  03:      4     Avg Loc [:3 HR AVG ]

06:     P80     Set Active Storage Area
  01:      1     Final Storage Area 1
  02:     25     Array ID or location

07:     P77     Real Time
  01:    110     Day,Hour-Minute

08:     P70     Sample
  01:      1     Reps
  02:      4     Loc 3 HR AVG

09:     P91     If Flag/Port
  01:     10     Do if flag 0 (output) is high
  02:     30     Then Do

10:     P54     Block Move
  01:      2     No. of Values
  02:      2     First Source Loc AVG i-1
  03:      1     Source Step
  04:      1     First Dest. Loc [:AVG i-2  ]
  05:      1     Destination Step

11:     P95     End

12:       P     End Table 1
```

8.2 RAINFALL INTENSITY

In this example, the total rain for the last 15 minutes is output only if any rain has occurred. The program makes use of the capability to direct the output of Output Processing Instructions to Input Storage.

Every 15 minutes, the total rain is sent to Input Storage. If the total is not equal to 0, output is redirected to Final Storage Area 1, the time is output and the total is sampled.

Input Location Labels:

1:Rain (mm)
2:15min tot

```
*          1          Table 1 Programs
  01:     60          Sec. Execution Interval

01:       P3          Pulse
  01:      1          Rep
  02:      1          Pulse Input Chan
  03:      2          Switch Closure
  04:      1          Loc [:Rain (mm)]
  05:   .254          Mult
  06:      0          Offset

02:      P92          If time is
  01:      0          minutes into a
  02:     15          minute interval
  03:     10          Set high Flag 0 (output)

03:      P80          Set Active Storage Area
  01:      3          Input Storage Area
  02:      2          Array ID or location

04:      P72          Totalize
  01:      1          Rep
  02:      1          Loc Rain (mm)

05:      P89          If X<=>F
  01:      2          X Loc 15min tot
  02:      2          <>
  03:      0          F
  04:     30          Then Do

06:      P80          Set Active Storage Area
  01:      1          Final Storage Area 1
  02:     25          Array ID or location

07:      P77          Real Time
  01:    110          Day,Hour-Minute

08:      P70          Sample
  01:      1          Reps
  02:      2          Loc 15min tot

09:      P95          End

10:        P          End Table 1
```

8.3 USING CONTROL PORTS AND LOOP TO RUN AM416 MULTIPLEXER

This example uses an AM416 to measure 16 copper-constantan thermocouples and 16 Model 223 soil moisture blocks. The sensors are read every ten minutes and the average value output once an hour. The multiplexer is housed in an AM-ENCT enclosure to minimize thermocouple errors created by thermal gradients. A 107 Temperature Probe is centrally located on the multiplexer board and used as a thermocouple temperature reference.

The AM416 switches the 223 moisture block out of the circuit when it is not being measured. This eliminates the need for the blocking capacitors used in the model 227 soil moisture block. The 223 blocks are about one fifth the cost of the 227 blocks.

Control ports are used to reset the AM416 and clock it through its channels. The sequence of the program is:

- Measure the 107 probe located at the AM416 for TC temperature reference.
- CR10 sets the port high which resets the AM416.
- A loop is entered; within each pass:
- The port clocking the AM416 is pulsed.
- The connected TCs and moisture blocks are measured.
- CR10 sets the port controlling AM416 reset low.
- Soil moisture measurements are converted to block resistances.

The input location in which the temperature and soil moisture measurements are stored is indexed to the loop counter (Instruction 87, Section 12). An indexed location is incremented by one with each pass through the loop. For example, on the first pass temperature is stored in Location 2, and soil moisture in Location 18. On the second pass temperature is stored in Location 3, and soil moisture in Location 19. After 16 loop passes, temperature and soil moisture measurements occupy Locations 2 through 17 and 18 through 33, respectively.

Connections are shown in Figure 8.3-1.

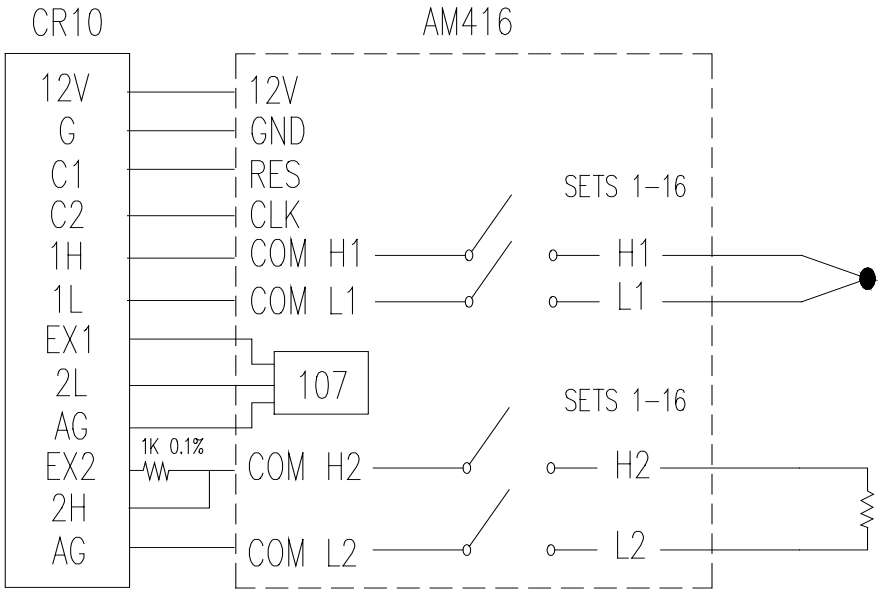


FIGURE 8.3-1. AM416 Wiring Diagram For Thermocouple and Soil Moisture Block Measurements

EXAMPLE PROGRAM MULTIPLEXING THERMOCOUPLES AND SOIL MOISTURE BLOCK

```
*          1      Table 1 Programs
  01:    600      Sec. Execution Interval

01:      P11      Temp 107 Probe
  01:      1      Rep
  02:      4      IN Chan
  03:      1      Excite all reps w/EXchan 1
  04:      1      Loc [:REF TEMP ]
  05:      1      Mult
  06:      0      Offset

02:      P86      Do
  01:     41      Set high Port 1

03:      P87      Beginning of Loop
  01:      0      Delay
  02:     16      Loop Count

04:      P86      Do
  01:     72      Pulse Port 2

05:      P14      Thermocouple Temp (DIFF)
  01:      1      Rep
  02:     21      2.5 mV 60 Hz rejection Range
  03:      1      IN Chan
  04:      1      Type T (Copper-Constantan)
  05:      1      Ref Temp Loc REF TEMP
  06:     2--     Loc [:TC TEMP#1]
  07:      1      Mult
  08:      0      Offset

06:      P5       AC Half Bridge
  01:      1      Rep
  02:     14      250 mV fast Range
  03:      3      IN Chan
  04:      1      Excite all reps w/EXchan 1
  05:    250      mV Excitation
  06:    18--     Loc [:SOIL M#1 ]
  07:      1      Mult
  08:      0      Offset

07:      P95      End

08:      P86      Do
  01:     51      Set low Port 1

09:      P59      BR Transform Rf[X/(1-X)]
  01:     16      Reps
  02:     18      Loc [:SOIL M#1 ]
  03:     .1      Multiplier (Rf)

10:      P92      If time is
  01:      0      minutes into a
  02:     60      minute interval
  03:     10      Set high Flag 0 (output)

11:      P77      Real Time
  01:    110      Day,Hour-Minute

12:      P71      Average
  01:     33      Reps
  02:      1      Loc REF TEMP
```

```
13:         P         End Table 1
```

8.4 SUB 1 MINUTE OUTPUT INTERVAL SYNCHED TO REAL TIME

Output can be synchronized to seconds by pressing "-" or "C" while entering the first parameter in Instruction 92. If a counter, incremented within the program, was used to determine when to set the Output Flag, output would depend on the number of times the table was executed. The actual time of output would depend on when the program was actually compiled and started running. If the table overran its execution interval (Section 1.1.1), the output interval would not be the count multiplied by the execution interval, but some longer interval.

In this example a temperature (107 Temperature Probe) is measured every 0.5 seconds and the average output every 30 seconds.

Input Location Assignments:

```
*           1         Table 1 Programs
  01         .5       Sec. Execution Interval

01:       P17         Module Temperature
  01:       1         Loc [:Ref_Temp ]

02:       P14         Thermocouple Tem (DIFF)
  01:       1         Rep
  02:      11         2.5 mV fast Range
  03:       2         IN Chan
  04:       2         Type E (Chromel-Constantan)

  05:       1         Ref Temp Loc Ref_Temp
  06:       2         Loc [:TC_Temp  ]
  07:       1         Mult
  08:       0         Offset

03:       P92         If time is
  01:       0--       minutes (seconds--) into a
  02:      30         minute or second interval
  03:      10         Set high Flag 0 (output)

04:       P71         Average
  01:       1         Rep
  02:       2         Loc TC_Temp

05:         P         End Table 1
```

8.5 INTERRUPT SUBROUTINE USED TO COUNT SWITCH CLOSURES (RAIN GAGE)

Subroutines given the label of 97 or 98 will be executed when control ports 7 or 8, respectively, go high (5 V, see Instruction 85, Section 12). In this example, Subroutine 98 and control port 8 are substituted for a pulse counting channel to count switch closures on a tipping bucket rain gage.

The subroutine adds 0.254 (mm, bucket calibrated for 0.01 inch tip) to an input location and uses Instruction 22 to delay 0.2 seconds.

The delay is to insure that any switch bouncing (when closing, the contacts actually bounce off each other, making and breaking the circuit several times) has died out before the subroutine is completed. (The pulse count inputs do this automatically.) Without the delay, the subroutine could be completed and called again by a bounce, causing false counts. The interrupt has no effect while the subroutine is still being executed.

Subroutine 98 is in effect keeping a running total in Input Storage. On the output interval, this total is sampled to Final Storage and zeroed by the program in Program Table 1.

An interrupt driven subroutine can interrupt a table while the Output Flag is set. The CR10 will complete whatever instruction it is executing, execute the subroutine, and then resume executing the table.

If the subroutine always added the count to the same location and a tip occurred while the total rain was being sampled, the subroutine would add the count to the input location before the location was zeroed, causing the count to be missed.

In the example, the subroutine adds the count to an alternate location when the Output Flag is set. Program Table 1 sets the Output Flag low after zeroing the location where the normal total is kept. The value in the alternate location is then added to location 12 and then the alternate location is zeroed.

To provide comparison, this example has the 2 pulse inputs also reading rain gages. (In a real

situation, it is more likely that the pulse counters would be used for 2 wind speeds.) In Program Table 1, the 2 normal pulse inputs are read and the hourly totals output to Final Storage with Instruction 72.

The rain gage is connected as diagrammed below. When the switch closes, 5 volts is applied to port 8 which causes the subroutine to be executed.

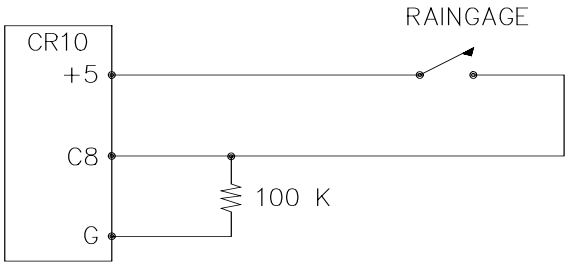


FIGURE 8.5-1. Connections for Rain Gage

Input location Assignments:

10:Rain #1	(from Pulse count)
11:Rain #2	(from Pulse count)
12:Rain #3	(from subroutine 98 while Output Flag is low)
13:Rain alt	(from subroutine 98 while Output Flag is high)

```
*          1        Table 1 Programs
  01:     10        Sec. Execution Interval

01:       P3        Pulse
  01:      2        Reps
  02:      1        Pulse Input Chan
  03:      2        Switch Closure
  04:     10        Loc [:Rain #1  ]
  05:   .254        Mult
  06:      0        Offset

02:      P92        If time is
  01:      0        minutes into a
  02:     60        minute interval
  03:     10        Set high Flag 0 (output)

03:      P77        Real Time
  01:    110        Day,Hour-Minute

04:        P        72  Totalize
  01:      2        Reps
  02:     10        Loc Rain #1

05:      P91        If Flag/Port
  01:     10        Do if flag 0 (output) is high
  02:     30        Then Do

06:      P70        Sample
  01:      1        Reps
  02:     12        Loc Rain #3

07:      P30        Z=F
  01:      0        F
  02:      0        Exponent of 10
  03:     12        Z Loc [:Rain #3  ]

08:      P86        Do
  01:     20        Set low Flag 0 (output)

09:      P33        Z=X+Y
  01:     13        X Loc Rain alt
  02:     12        Y Loc Rain #3
  03:     12        Z Loc [:Rain #3  ]

10:      P30        Z=F
  01:      0        F
  02:      0        Exponent of 10
  03:     13        Z Loc [:Rain alt ]

11:      P95        End

12:        P        End Table 1

*          3        Table 3 Subroutines

01:      P85        Beginning of Subroutine
  01:     98        Subroutine Number

02:      P91        If Flag/Port
  01:     10        Do if flag 0 (output) is high
  02:     30        Then Do

03:      P34        Z=X+F
  01:     13        X Loc Rain alt
  02:  0.254        F
  03:     13        Z Loc [:Rain alt ]

04:      P94        Else

05:      P34        Z=X+F
  01:     12        X Loc Rain #3
  02:   .254        F
  03:     12        Z Loc [:Rain #3  ]

06:      P95        End

07:      P22        Excitation with Delay
  01:      1        EX Chan
  02:      0        Delay w/EX (units=.01sec)
  03:     20        Delay after EX (units=.01sec)
  04:      0        mV Excitation

08:      P95        End

09:        P        End Table 3
```

8.6 SDM-A04 ANALOG OUTPUT MULTIPLEXER TO STRIP CHART

This example illustrates the use of the SDM-A04 4 Channel Analog Output Multiplexer to output 4 analog voltages to a strip chart.

While of questionable value because of current requirements and strip chart reliability, some archaic regulations require strip chart backup on weather data. The SDM-A04 may be used with the CR10 to provide analog outputs to strip charts. The output values in this example are wind speed, wind direction, air temperature, and solar radiation.

Instruction 103 is used to activate the SDM-A04. The (4) millivolt values to output must be stored in adjacent Input Storage locations, the first of which is referenced in Instruction 103.

The following program measures the sensors every 5 seconds. The readings are moved to another 4 locations and scaled to a 0 to 1000 millivolt output for the SDM-A04. Wind direction is changed from a 0-360 degree input to output representing 0 to 540 degrees. This conversion is done in a subroutine which is described in the next example.

The example also includes instructions to output wind vector and average temperature and solar radiation every hour.

Input Location Labels:

```
1:WS
2:0-360 WD
3:Ta
4:SR
5:WS output
6:0-540 out
7:Ta output
8:SR output
10:0-540 WD
```

```
*          1          Table 1 Programs
  01:      5          Sec. Execution Interval

01:        P3         Pulse
  01:      1          Rep
  02:      1          Pulse Input Chan
  03:     22          Switch Closure
  04:      1          Loc [:WS      ]
  05:      1.789      Mult
  06:      1          Offset

02:        P4         Excite,Delay,Volt(SE)
  01:      1          Rep
  02:     14          250 mV fast Range
  03:      1          IN Chan
  04:      1          Excite all reps w/EXchan 1
  05:      5          Delay (units .01sec)
  06:   1000          mV Excitation
  07:      2          Loc [:0-360 WD ]
  08:       .7273     Mult
  09:      0          Offset

03:        P11        Temp 107 Probe
  01:      1          Rep
  02:      2          IN Chan
  03:      2          Excite all reps w/EXchan 2
  04:      3          Loc [:Ta      ]
  05:      1.8        Mult
  06:     32          Offset

04:        P1         Volt (SE)
  01:      1          Rep
  02:      2          7.5 mV slow Range
  03:      3          IN Chan
  04:      4          Loc [:SR      ]
  05:       .14493    Mult
  06:      0          Offset

05:        P92        If time is
  01:      0          minutes into a
  02:     60          minute interval
  03:     10          Set high Flag 0 (output)

06:        P54        Block Move
  01:      4          No. of Values
  02:      1          First Source Loc WS
  03:      1          Source Step
  04:      5          First Dest. Loc [:WS output]
  05:      1          Destination Step

07:        P86        Do
  01:      1          Call Subroutine 1

08:        P53        Scaling Array (A*loc +B)
  01:      5          Start Loc [:WS output]
  02:     10          A1 SCALE WS,
                      0-100 MPH = 0-1000 MV
  03:      0          B1
  04:      1.8519     A2 SCALE WD,
                      0-540 DEG = 0-1000 MV
  05:      0          B2
  06:      5.7143     A3 SCALE TEMP,
                      -25 - 100 F =0-1000 MV
  07:     25          B3
  08:   1000          A4 SCALE RADIATION,
                      0-1KW/M^2 = 0-1000 MV
  09:      0          B4
```

```
09:      P      103 SDM-A04
  01:    4      Reps
  02:    30     Address
  03:    5      Loc WS output

10:      P92    If time is
  01:    0      minutes into a
  02:    60     minute interval
  03:    10     Set high Flag 0 (output)

11:      P69    Wind Vector
  01:    1      Rep
  02:    180    Samples per sub-interval
  03:    00     Polar Sensor/(S ,D1, SD1)
  04:    1      Wind Speed/East Loc WS
  05:    2      Wind Direction/North Loc
                0-360 WD

12:      P71    Average
  01:    2      Reps
  02:    3      Loc Ta

13:      P      End Table 1
```

8.7 CONVERTING 0-360 WIND DIRECTION OUTPUT TO 0-540 FOR STRIP CHART

If 0-360 degree wind direction is output to a strip chart the discontinuity at 0/360 will cause the pen to jump back and forth full scale when the winds are varying from the north. In the days of strip charts this was solved with a 0-540 degree pot on the wind vane (direction changes from 540 to 180 and from 0 to 360 so the pen only jumps once when the wind is out of the north or south).

When faced with the necessity of strip chart output (see previous example), the following algorithm can be used to change a 0-360 degree input to 0-540. (If you have a 0-540 pot, it can be used with the CR10 since the Wind Vector Instruction, 69, will work with this output.)

To change 0-360 degrees to the 0-540 degrees, 360 degrees must sometimes be added to the reading when it is in the range of 0 to 180. The following algorithm does this by assuming that if the previous reading was less than 270, the vane has shifted through 180 degrees and does not need to be altered. If the previous 0-540 reading was greater than 270, 360 degrees is added.

This example is written as a subroutine which is used by the previous example to output an analog voltage to a strip chart.

Input Location Labels:

```
 2:0-360 WD
 6:0-540 out
10:0-540 WD
```

```
*        3      Table 3 Subroutines
01:      P85    Beginning of Subroutine
  01:    1      Subroutine Number

02:      P89    If X<=>F
  01:    10     X Loc 0-540 WD
  02:    3      >=
  03:    270    F
  04:    30     Then Do

03:      P86    Do
  01:    11     Set high Flag 1

04:      P94    Else

05:      P86    Do
  01:    21     Set low Flag 1

06:      P95    End

07:      P31    Z=X
  01:    2      X Loc 0-360 WD
  02:    10     Z Loc [:0-540 WD]

08:      P89    If X<=>F
  01:    10     X Loc 0-540 WD
  02:    4      <
  03:    180    F
  04:    30     Then Do

09:      P91    If Flag/Port
  01:    11     Do if flag 1 is high
  02:    30     Then Do

10:      P34    Z=X+F
  01:    10     X Loc 0-540 WD
  02:    360    F
  03:    10     Z Loc [:0-540 WD]

11:      P31    Z=X
  01:    10     X Loc 0-540 WD
  02:    6      Z Loc [:0-540 out]

12:      P95    End

13:      P95    End
```

```
14:     P95         End

15:       P         End Table 3
```

8.8 USE OF 2 FINAL STORAGE AREAS - SAVING DATA PRIOR TO EVENT

One of the uses of 2 Final Storage Areas is to save a fixed amount of data before and after some event.

In this example, a load cell is measured every second. It is assumed that at some random interval the load will exceed 25 pounds for less than 10 seconds. Exceeding 25 pounds is the event to be captured. The data from the 10 seconds before the event and 10 seconds after the event is to be saved (21 seconds including the scan in which the load first exceeds 25 pounds).

Every second the load cell is measured; hours-minutes, seconds, and the load are output to Final Storage Area 2 (4 values with the Array ID). 84 locations are allocated to Final Storage Area 2. Thus, Area 2 holds 21 seconds (4 values/second x 21 seconds = 84 locations).

When 25 pounds is exceeded, 10 is loaded into an input location and flag 1 is set high. The input location is used as a down counter. The flag indicates an event has occurred and prevents the input location from being reloaded until 11 seconds have passed.

The down counter is decremented by 1 each time the table is executed. When it equals 0 all the data in Final Storage Area 2 is transferred to Final Storage Area 1 (using Instruction 96) and Flag 1 is set low.

The down counter is set to 10 instead of 11 because it is decremented after checking to see if it is 0.

Input Location Labels:

1:FORCE LBS
2:DOWN CNT

```
*          1        Table 1 Programs
  01:      1        Sec. Execution Interval

01:       P6        Full Bridge
  01:      1        Rep
  02:     22        7.5 mV 60 Hz rejection Range

  03:      1        IN Chan
  04:      1        Excite all reps w/EXchan 1
  05:   2500        mV Excitation
  06:      1        Loc [:FORCE LBS]
  07:   33.333      Mult
  08:      0        Offset

02:      P86        Do
  01:     10        Set high Flag 0 (output)

03:      P80        Set Active Storage Area
  01:      2        Final Storage Area 2
  02:     10        Array ID or location

04:      P77        Real Time
  01:     11        Hour-Minute,Seconds

05:      P70        Sample
  01:      1        Reps
  02:      1        Loc FORCE LBS

06:      P89        If X<=>F
  01:      1        X Loc FORCE LBS
  02:      3        >=
  03:     25        F
  04:     30        Then Do

07:      P91        If Flag/Port
  01:     21        Do if flag 1 is low
  02:     30        Then Do

08:      P86        Do
  01:     11        Set high Flag 1

09:      P30        Z=F
  01:     10        F
  02:      0        Exponent of 10
  03:      2        Z Loc [:DOWN CNT ]

10:      P95        End

11:      P95        End

12:      P89        If X<=>F
  01:      2        X Loc DOWN CNT
  02:      1        =
  03:      0        F
  04:     30        Then Do

13:      P91        If Flag/Port
  01:     11        Do if flag 1 is high
  02:     30        Then Do

14:      P96        Serial Output
  01:     81        All data to other FS Area

15:      P86        Do
  01:     21        Set low Flag 1

16:      P95        End
```

```
17:     P94         Else

18:     P34         Z=X+F
  01:     2         X Loc DOWN CNT
  02:    -1         F
  03:     2         Z Loc [:DOWN CNT ]

19:     P95         End

20:       P         End Table 1

*         A         Mode 10 Memory
                    Allocation
  01:    28         Input Locations
  02:    64         Intermediate Locations
  03:    84         Final Storage Area 2
```

8.9 LOGARITHMIC SAMPLING USING LOOPS

A ground water pump test requires that water level be measured and recorded according to the following schedule.

Time into Test, min			Output Interval Loop #
00	to 10	10 sec.	1
10	to 30	30 sec.	2
30	to 100	1 min.	3
100	to 300	2 min.	4
300	to 1000	5 min.	5
1000	and greater	10 min.	6

This is accomplished with a series of loops (Instruction 87), where the delay and count parameters are used to implement the frequency of measurement (and output) and the duration of the that frequency. The unit of delay is the execution interval. A delay of 1 with a 10 second execution interval and a count of 60 means the instructions in the loop, in this case measure and output water level, are executed every 10 seconds for 10 minutes.

The drawdown portion of the test is completed at some time greater than 1000 minutes. To enter the recharge phase of the test, the operator enters the *6AD Mode and sets Flag 1 high. At the next 10 minute pass through loop 6 the loop is exited. Program execution returns to the top of the program table and the measurement schedule starts over again for the recharge test.

The sensor is a 50 PSI Druck, model 930/ti with a calibration of 49.93 mV/10V of excitation or 4.993mV/V. Your calibration will be different.

An excitation voltage of 1500 mV yields a maximum signal of 7.489 mV at 50 PSI, fully utilizing the 7.5 mV Input Range to provide the best resolution.

The multiplier, m, is calculated to provide depth of water in feet:

m = (50 psi/4.993 mV/V) * (2.3067 ft/psi)

m = 23.099 ft/mV/V

The offset is calculated to provide a final value that represents the distance from the lip of the well to the water surface. Similar to Figure 7.16-2, the offset equals the initial distance of 47.23 feet plus the initial reading of 54.77, or 102 feet.

```
*         1         Table 1 Programs
  01:    10         Sec. Execution Interval
```

User must toggle flag 1 to start measurements

```
01:     P91         If Flag/Port
  01:    21         Do if flag 1 is low
  02:     0         Go to end of program table
```

Loop 1, Output every 10 seconds for 10 minutes

```
02:     P87         Beginning of Loop
  01:     1         Delay
  02:    60         Loop Count

03:     P86         Do
  01:     1         Call Subroutine 1

04:     P95         End
```

Loop 2, Output every 30 seconds for 20 minutes

```
05:     P87         Beginning of Loop
  01:     3         Delay
  02:    40         Loop Count

06:     P86         Do
  01:     1         Call Subroutine 1

07:     P95         End
```

Loop 3, Output every 1 minute for 70 minutes

```
08:     P87         Beginning of Loop
  01:     6         Delay
  02:    70         Loop Count

09:     P86         Do
```

```
  01:      1          Call Subroutine 1

10:      P95          End
```

Loop 4, Output every 2 minutes for 200 minutes

```
11:      P87          Beginning of Loop
  01:     12          Delay
  02:    100          Loop Count
```

```
12:     P86         Do
  01:     1         Call Subroutine 1

13:     P95         End
```

Loop 5, Output every 5 minutes for 700 minutes

```
14:     P87         Beginning of Loop
  01:    30         Delay
  02:   140         Loop Count

15:     P86         Do
  01:     1         Call Subroutine 1

16:     P95         End
```

Loop 6, Output every 10 minutes until stopped by user

```
17:     P87         Beginning of Loop
  01:    60         Delay
  02:     0         Loop Count

18:     P86         Do
  01:     1         Call Subroutine 1

19:     P91         If Flag/Port
  01:    21         Do if flag 1 is low
  02:    31         Exit Loop if true

20:     P95         End

21:       P         End Table 1

*         3         Table 3 Subroutines

01:     P85         Beginning of Subroutine
  01:     1         Subroutine Number

02:      P6         Full Bridge
  01:     1         Rep
  02:    22         7.5 mV 60 Hz rejection Range

  03:     1         IN Chan
  04:     1         Excite all reps w/EXchan 1
  05:  1500         mV Excitation
  06:     1         Loc [:LEVEL FT.]
  07:    .46199     Mult
  08:   102         Offset

03:     P86         Do
  01:    10         Set high Flag 0 (output)

04:     P77         Real Time
  01:   111         Day,Hour-Minute,Seconds

05:     P70         Sample
  01:     1         Reps
  02:     1         Loc LEVEL FT.

06:     P95         End

07:       P         End Table 3
```

SECTION 9. INPUT/OUTPUT INSTRUCTIONS

TABLE 9-1. Input Voltage Ranges and Codes

Range Code				Full Scale Range	Resolution*
Slow 2.72ms Integ.	Fast 250µs Integ.	60 Hz Reject	50 Hz Reject		
1	11	21	31	±2.5 mV	0.33 µV
2	12	22	32	±7.5 mV	1. µV
3	13	23	33	±25 mV	3.33 µV
4	14	24	34	±250 mV	33.3 µV
5	15	25	35	±2500 mV	333. µV

* *Differential measurement; resolution for single-ended measurement is twice value shown.*

NOTE: When a voltage input exceeds the range programmed, the value which is stored is set to the maximum negative number and displayed as -99999 in high resolution or -6999 in low resolution.

*** 1 SINGLE-ENDED VOLTS ***

FUNCTION
This Instruction is used to measure voltage at a single-ended input with respect to ground. Output is in millivolts.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Single-ended channel number for first measurement
04:	4	input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 per measurement

*** 2 DIFFERENTIAL VOLTS ***

FUNCTION
This Instruction reads the voltage difference between the high and low inputs of a differential channel. Table 9-1 contains all valid voltage ranges and their codes. Both the high and low inputs must be within ±2.5 V of the CR10's ground (see Common Mode Range in Section 14.3.2). Pyranometer and thermopile sensors require a jumper between LO and GROUND to keep them in Common Mode Range. Output is in millivolts.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Differential channel number for first measurement
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 per measurement

*** 3 PULSE COUNT ***

There are three pulse input types which may be measured with the Pulse Count Instruction.

HIGH FREQUENCY PULSE

In this configuration, the minimum detectable pulse width is 2 microseconds, i.e. the maximum frequency is 250 kHz with a 50% duty cycle. The 8 bit counter has a maximum input frequency of 2000 Hz and the 16 bit option a maximum of 250 kHz. The count is incremented when input voltage changes from below 1.5 volts to above 3.5 volts. The

maximum input voltage is +20 volts. A problem, however, arises when the pulse is actually a low frequency signal (below about 10 Hz) and the positive voltage excursion exceeds 5.6 VDC.

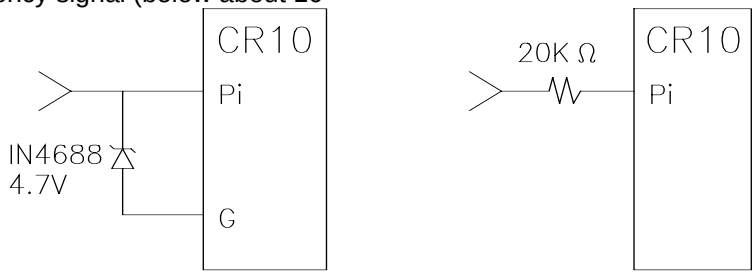


FIGURE 9-1. Conditioning for Long Duration Voltage Pulses

When this happens, the excess voltage is shunted to the CR10 5 VDC supply, with the current limited by an internal 10 Kohm resistor. When this extra current source exceeds the quiescent current needs of the CR10 (about 0.6 mA), the 5 VDC supply will start to rise, upsetting all analog measurements.

Thus, pulses whose positive voltage portion exceeds 5.6 VDC with a duration longer than 100 milliseconds need external conditioning. One method would be to use a 4 to 5.6 V zener diode from the signal to ground. The simplest method, however, is to add an external 20 Kohm resistor in series with the signal (Figure 9-1). This will limit the current for pulses to 20 VDC to the point that it will not upset the CR10 5 VDC supply.

LOW LEVEL AC

This configuration is used to count the frequency of AC signals from magnetic pulse flow transducers or other low voltage, sine wave inputs. The minimum input voltage is 6 millivolts RMS. Input hysteresis is 11 millivolts. The maximum AC input voltage is 20 volts RMS. The maximum input frequency ranges from 100 Hz at 20 mV RMS to 1000 Hz at 150 mV or greater. Consult the factory if higher frequencies are desired.

SWITCH CLOSURE

In this configuration, the minimum switch closed time is 5 milliseconds. The minimum switch open time is 6 milliseconds. The maximum bounce time is 1 millisecond open without being counted.

The 2 pulse count input channels each have eight bit counters. Input frequencies greater than 2000 Hz (the limit of the eight bit counter, 255 counts at the reset interval of 0.125 second) can be counted by combining two counters on one input channel. When this option is selected, channel 1 is used for the pulse input. Channel 2 is not used.

Every 0.125 seconds, the CR10 processor transfers the values from the 8 bit pulse counters into 16 bit accumulators (max count is 65,535) and the 8 bit counters are hardware reset to zero. The pulses accumulate in these 16 bit accumulators until the program table containing the Pulse Count Instruction is executed. At the beginning of the execution of the Table containing the Pulse Count Instruction, the total in the 16 bit accumulator is transferred to a temporary RAM buffer. The 16 bit accumulator is then zeroed. When the table execution reaches the Pulse Count Instruction, the value in the RAM buffer is multiplied by the multiplier and added to the offset and placed into the designated input location.

> **CAUTION:** The RAM buffer does NOT accumulate counts; it is zeroed each time the table is executed regardless of whether or not the pulse instruction is executed. If all counts are necessary, it is imperative that the Pulse Count Instruction be executed (not branched around) every time the table is executed.

If a table execution was skipped because the processor was executing the previous table (Section 2.1) or if the user resets the time, the value in the 16 bit accumulator is the result of a longer than normal interval. This value can either be used or it can be discarded. If pulse counts are being totalized, a missing count could be significant and the value from the erroneously long interval should NOT be discarded. If the pulse count is being processed in a way in which the resultant value

is dependent upon the sampling interval (e.g., speed, RPM), the value from the excessive interval should be discarded. If the value is discarded the value in the RAM buffer from the previous measurement will be used.

There is also an option to output the count as a frequency (i.e., counts/execution interval in seconds = Hz) as well as discard the result from an excessive interval. This allows the use of a conversion factor that is independent of the execution interval.

The options of discarding counts from long intervals, pulse input type, and using a 16 bit counter are selected by the code entered for the 4th parameter (Table 9-2).

> **NOTE:** All pulse count instructions must be kept in the same table. If the Pulse Count Instruction is contained within a subroutine, that subroutine must be called from Table 2.

TABLE 9-2. Pulse Count Configuration Codes

Code	Configuration
0	High frequency pulse
1	Low level AC
2	Switch closure
3	High frequency pulse, sixteen bit counter
4	Low level AC, sixteen bit counter
1X	Long interval data discarded
2X	Long interval data discarded, frequency (Hz) output

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Pulse channel number for first measurement
03:	2	Configuration code (from above table)
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 per measurement

Intermediate storage locations altered: 1 for each repetition

*** 4 EXCITE, DELAY, AND MEASURE ***

FUNCTION
This instruction is used to apply an excitation voltage, delay a specified time, and then make a single-ended voltage measurement. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Single-ended channel number for first measurement
04:	2	Excitation channel number
05:	4	Delay in hundredths of a second
06:	4	Excitation voltage (millivolts)
07:	4	Input location number for first measurement
08:	FP	Multiplier
09:	FP	Offset

Input locations altered: 1 per measurement

*** 5 AC HALF BRIDGE ***

FUNCTION
This instruction is used to apply an excitation voltage to a half bridge (Figure 13.5-1), make a single-ended voltage measurement of the bridge output, reverse the excitation voltage, then repeat the measurement. The difference between the two measurements is used to calculate the resulting value which is the ratio of the measurement to the excitation voltage. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

The excitation "on time" for each polarity is exactly the same to insure that ionic sensors do not polarize with repetitive measurements. The range should be selected to be a fast measurement (range 11-15), limiting the excitation on time to less than 800 microseconds at each polarity. A slow integration time should not be used with ionic sensors because of polarization error.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range Code (Table 9-1)
03:	2	Single-ended channel number
04:	2	Excitation channel number
05:	4	Excitation voltage (millivolts)
06:	4	Input location number for first measurement
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 per measurement

*** 6 FULL BRIDGE WITH SINGLE DIFFERENTIAL MEASUREMENT ***

FUNCTION
This Instruction is used to apply an excitation voltage to a full bridge and make a differential voltage measurement of the bridge output. The measurement is made with the polarity of the excitation voltage both positive and negative (Figure 13.5-1). The result is 1000 times the ratio of the measurement to the excitation voltage. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Differential channel number for first measurement
04:	2	Excitation channel number
05:	4	Excitation voltage (millivolts)
06:	4	Input location number for first measurement
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 per measurement

*** 7 THREE WIRE HALF BRIDGE ***

FUNCTION
This Instruction is used to determine the ratio of the sensor resistance to a known resistance using a second voltage sensing wire from the sensor to compensate for lead wire resistance.

The measurement sequence is to apply an excitation voltage, make two voltage measurements on two adjacent single-ended channels, the first on the reference resistor and the second on the voltage sensing wire from the sensor (Figure 13.5-1), then reverse the excitation voltage and repeat the measurements. The two measurements are used to calculate the resulting value, which is the ratio of the voltage across the sensor to the voltage across the reference resistor. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code for both measurements (Table 9-1)
03:	2	Single-ended channel number for first measurement
04:	2	Excitation channel
05:	4	Excitation voltage (millivolts)
06:	4	Input location number for first measurement
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 per measurement

*** 8 DIFFERENTIAL VOLTAGE WITH EXCITATION AND DELAY ***

FUNCTION
This measurement consists of applying a single excitation voltage, delaying a specified time, and making a differential voltage measurement. The result stored is the voltage measured.

"Delay" (Parameter 5) refers to increasing the signal settling time by increasing the time between the start of excitation and the start of signal integration (Section 13.2). If a delay of 0

is specified, the inputs for the differential measurement are not switched for a second integration as is normally the case. With the 0 delay, Instruction 8 does not have as good resolution or common mode rejection as other differential measurements. It does provide a very rapid means of making bridge measurements. This instruction does not reverse excitation. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Differential channel number for first measurement
04:	2	Excitation channel number
05:	4	Delay (0.01s)
06:	4	Excitation voltage (millivolts)
07:	4	Input location number for first measurement
08:	FP	Multiplier
09:	FP	Offset

Input locations altered: 1 per measurement

*** 9 FULL BRIDGE WITH EXCITATION *** COMPENSATION

FUNCTION
This instruction is used to apply an excitation voltage and make two differential voltage measurements. The measurements are made with both positive and negative excitation voltage. The measurements are made on sequential channels. The result is the voltage measured on the second channel (V_2) divided by the voltage measured on the first (V_1). If V_1 is measured on the 2.5 V range (code 5,15, 25 or 35 in Parameter 2), then the result is 1000 times V_2/V_1. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

When used as a 6 wire full bridge (Figure 13.5-1), the connections are made so that V_1 is the measurement of the voltage drop across the full bridge, and V_2 is the measurement of the bridge output. Because the excitation voltage for a full bridge measurement is usually in the 2.5 V range, the output is usually 1000 V_2/V_1 or millivolts output per volt excitation.

When used to measure a 4 wire half bridge, the connections are made so that V_1 is the voltage drop across the fixed resistor (R_f), and V_2 is the drop across the sensor (R_s). As long as V_1 is not measured on the 2.5V range, the result is V_2/V_1 which equals R_s/R_f.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code for V_1 (Table 9-1)
03:	2	Range code for V_2
04:	2	Differential channel number for first measurement
05:	2	Excitation channel number
06:	4	Excitation voltage (millivolts)
07:	4	Input location number for first measurement
08:	FP	Multiplier
09:	FP	Offset

Input locations altered: 1 per measurement

*** 10 BATTERY VOLTAGE ***

FUNCTION
This instruction reads the battery voltage and writes it to an input location. The units for battery voltage are volts. When the batteries are around 8 V, false battery readings of 9 to 10 V will result, and the quiescent current drain increases from 0.7 mA to 7 mA. At 9.2 to 9.3 V, false analog measurements are possible (Example: 2000 mV input is measured as 2010 to 2050 mV).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location

Input locations altered: 1

*** 11 107 THERMISTOR PROBE ***

FUNCTION
This Instruction applies a 2 VAC excitation voltage to Campbell Scientific's Model 107

Thermistor Probe, makes a fast, single-ended voltage measurement across a resistor in series with the thermistor, and calculates the temperature in °C with a polynomial. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition. The maximum polynomial error from -40°C to +56°C is given here:

Curve Fit Error --

Range (°C)	Error (°C)
-40 to +56	+1.0
-24 to +48	+0.1

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Single-ended channel number of first measurement
03:	2	Excitation channel number
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 for each thermistor channel

***** 12 207 RELATIVE HUMIDITY PROBE *****

FUNCTION
This instruction applies a 1.5 VAC excitation across Campbell Scientific's Model 207 Temperature and RH Probe, makes a fast single-ended measurement across a series resistor, calculates the result with a 5th order polynomial, and performs the required temperature compensation before outputting the result in % RH.

When measuring several probes, all the RH elements should be connected sequentially. The temperature values used to correct the RH measurements should also be stored sequentially to make use of the REP feature in Instruction 11.

NOTE: The temperature value used in compensating the RH value (Parameter 5) must be obtained (see Instruction 11) prior to executing Instruction 12 and must be in Celsius.

The RH results are placed sequentially into the input locations beginning with the first RH value.

In the 207 probe, the RH and temperature elements use a common excitation line.

CAUTION: Never excite the 207 probe with DC excitation because the RH chip will be damaged.

A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

The maximum RH polynomial error is given here:

Curve Fit Error --

Range (%RH)	Error (%RH)
10 - 100	+4
15 - 94	+1

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	First single-ended channel for RH measurement
03:	2	Excitation channel number
04:	4	Input location for first compensating temperature measurement (°C)
05:	4	Input location for first R.H. measurement
06:	FP	Multiplier
07:	FP	Offset

Input locations altered: 1 for each RH measurement

***** 13 THERMOCOUPLE TEMPERATURE, *****
SINGLE-ENDED

FUNCTION
This instruction uses the selected thermocouple calibration to calculate the thermocouple output voltage at the reference temperature, then it makes a SINGLE-ENDED VOLTAGE MEASUREMENT (Section 13.2) on the thermocouple and adds the measured voltage

to the calculated reference voltage, then	converts the voltage to temperature in °C.

TABLE 9-3. Thermocouple Type Codes

Code	Thermocouple Type
X1	T (copper - constantan)
X2	E (chromel - constantan)
X3	K (chromel - alumel)
X4	J (iron - constantan)

X = 0	Normal Measurement
X = 8	TC input from A5B40 isolation (uses 5 V range)
X = 9	Output -99999 if out of common mode range (Inst. 14 only)

TABLE 9-4. Voltage and Temperature Ranges for Thermocouples if the Reference is 20°C

Voltage Range	Type T	Type E	Type K	Type J
±2.5 mV	-46 to 50	-21 to 60	-44 to 80	-29 to 67
±7.5 mV	-270 to 181	-124 to 134	-270 to 620	-149 to 159
±25 mV	-270 to 400	-270 to 365	-270 to 620	-210 to 476
±250 mV	–––	-270 to 1000	-270 to 1372	-210 to 760

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-4)
03:	2	Single-ended channel number for first TC
04:	2	Thermocouple type code
05:	4	Reference temperature location
06:	4	Destination input location
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 for each thermocouple channel

*** 14 THERMOCOUPLE TEMPERATURE, *** DIFFERENTIAL MEASUREMENT

FUNCTION
This instruction calculates the thermocouple temperature for the thermocouple type selected. The instruction specifies a DIFFERENTIAL VOLTAGE MEASUREMENT (Section 13.2) on the thermocouple, adds the measured voltage to the voltage calculated for the reference temperature relative to 0°C, and converts the combined voltage to temperature in °C. The differential inputs are briefly shorted to ground prior to making the voltage measurement to insure that they are within the common mode range. Table 9-3 gives the thermocouple type codes for Parameter 4. Enter a 9 in front of the code and the CR10 will make an additional check on common mode range; -99999 is output for temperature if common mode range is exceeded.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Differential channel number for first TC
04:	2	Thermocouple type code
05:	4	Reference temperature location
06:	4	Destination input location
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 for each thermocouple channel

*** 16 TEMPERATURE FROM *** PLATINUM R.T.D.

FUNCTION
This instruction uses the result of a previous RTD bridge measurement to calculate the temperature according to the DIN 43760 specification adjusted (1980) to conform to the International Electrotechnical Commission standard. The range of linearization is -200° to 850°C. The error in the linearization is less than 0.001°C between -100° and +300°C, and is less than 0.003°C between -180° and +830°C. The error (T calculated - T standard)

is +0.006° at -200°C and -0.006° at +850°C. The input must be the ratio Rs/Ro, where Rs is the RTD resistance and Ro the resistance of the RTD at 0°C (Sections 7.9 and 7.10).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Input location of Rs/Ro
03:	4	Input location of result
04:	FP	Multiplier
05:	FP	Offset

Input locations altered: 1 for each RTD

*** 17 INTERNAL TEMPERATURE ***

FUNCTION
This instruction measures the temperature (°C) of a thermistor on the CR10 analog board. This temperature is generally not adequate for use as the thermocouple reference temperature (Section 13.4).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location number for temperature

Input locations altered: 1

*** 18 MOVE TIME TO INPUT LOCATION ***

FUNCTION
This instruction takes the current time in seconds into the minute, minutes into the day, or hours into the year and does a modulo divide (see Instruction 46) on the time value with the number specified in the second parameter. The result is stored in the specified input location. Entering 0 or a number which is greater than the maximum value of the time for the second parameter will result in the actual time value being stored.

PARAMETER 1 CODES

Code	Time Units
0	Seconds into minute (maximum 60)
1	Minutes into current day (maximum 1440)
2	Hours into current year (maximum 8784)

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Time Code
02:	4	Number to modulo divide by
03:	4	Input location number

Input locations altered: 1

*** 19 MOVE SIGNATURE INTO INPUT LOCATION ***

FUNCTION
This instruction stores the signature of the Read Only Memory (ROM) and user program memory (RAM) into an input location. The signature is a result of the CR10 PROM, the size of RAM, and the entries in the *1, *2, *3, *A, and *C Modes. This signature is not the same as the signatures given in the *B Mode. Recording the signature allows detection of any program change or ROM failure.

PARAM NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location number

Input locations altered: 1

*** 20 PORT SET ***

FUNCTION
This instruction sets or configures specified control ports (C1-C8). On power-up, ports default to input configuration (i.e., they are not driven high or low by the CR10, and can be used to read the status of an external signal using Instruction 25). When a port is set high, low, pulsed, or toggled by this instruction or a program control command, the port is automatically configured as an output.

> **NOTE**: Voltages in excess of 5.5 volts applied to a control port can cause the CR10 to malfunction.

Ports can also be set using the *6 Mode or the J and K telecommunications commands. However, the ports MUST be configured as outputs before these means of setting them will work. The option to configure the port as an output is used when a port must be configured as an output without changing the state of the port.

Pulse duration, initiated by a program control instruction, can be set for each control port (Table 12-2). Instruction 20 does not pulse the port, it only sets the duration. If Instruction 20 is not used to set the duration, the pulse command will result in a 10 ms pulse.

Instruction 20 has two 4 digit parameters. Each digit represents one control port. The code (0-9) entered as the digit determines what effect command 20 has on the corresponding port.

TABLE 9-5. Port Configuration Option Codes

Code	Function
0	Set port low
1	Set port high
2	Toggle port
3	Pulse duration 1 ms
4	Pulse duration 10 ms
5	Pulse duration 100 ms
6	Pulse duration 1 s
7	Configure as output
8	Configure as input
9	Leave unchanged

Duration of pulse on subsequent pulse port command in Program Control Instruction.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	C8, C7, C6, C5 option codes
02:	4	C4, C3, C2, C1 option codes

Input locations altered: 0

*** 22 EXCITATION WITH DELAY ***

FUNCTION
This instruction is used in conjunction with others for measuring a response to a timed excitation using the switched analog outputs. It sets the selected excitation output to a specific value, waits for the specified time, then turns off the excitation and waits an additional specified time before continuing execution of the following instruction. Analog power is turned off during delay after excitation to drop power to 3 mA.

If the only requirement is the delay of program execution, the excitation on time (parameter 2) can be set to zero and the off time delay (parameter 3) can be used.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Excitation channel number
02:	4	Delay time in hundredths of a second that excitation is on
03:	4	Delay time in hundredths of a second after excitation is turned off
04:	FP	Excitation voltage in millivolts

Input locations altered: 0

*** 23 BURST MEASUREMENT ***

FUNCTION
Instruction 23 will repeatedly make a voltage measurement on a series of single-ended or differential channels, applying excitation if desired. The measurement units are millivolts prior to scaling. The measurements saved can be those made immediately upon execution of the instruction or grouped around a specified trigger condition. The results of the measurements may be stored in Input Storage or the raw A/D data can be sent out the serial port. The minimum sample time per channel is 1.333 ms (i.e., one channel can be sampled at a maximum rate of 750 Hz).

MEASUREMENT
The voltage measurement must use the fast integration time of 272 µs. Differential measurements are made with a single integration. The noise level on a fast single-ended or a differential measurement with 1 integration is 3 µV RMS. Thus, only the 25 mV and greater voltage ranges are practical (range codes 13-15).

Excitation is always supplied from excitation channel 1. The excitation voltage in millivolts is entered in Parameter 9. If excitation is not desired, enter 0 for Parameter 9.

Three options are available for the first digit in Parameter 4.

0 - Trigger on first measurement channel.
1 - Trigger on Digital Control Port 1.
2 - Trigger on first measurement channel, set Digital Control Port high when trigger is met and low when measurements have finished.

When triggering on options 0 or 2, the measurement on the first specified channel (Parameter 3) is compared to the limit specified in Parameter 8. The user's multiplier and offset are not applied before the comparison: the limit must be entered in units of millivolts. If a digital trigger (low < 1.5V, 3.5 V < HIGH < 5 V) is used, it must be input to Digital Control Port #1. Option 2 is useful when 2 or more CR10's are required to start "Bursting" at the same time. For example, Digital Control Port #1 of all the CR10's involved are connected in parallel. The master CR10 sets its Digital Control Port #1 high when the trigger condition is met. The remaining dataloggers, programmed with option 1, trigger on the digital signal. The master CR10 sets Port #1 low when its measurements are completed.

Five trigger options are available for the second digit in Parameter 4: **0** trigger immediately, start saving or sending data with the first measurement made; **1** trigger if the measurement is greater than the limit entered or if the digital trigger is high; **2** trigger if the measurement is less than the limit or if the digital trigger is low; **3** trigger on rising edge; i.e., when the measurement goes from below to above the limit or when the digital trigger goes from low to high; **4** trigger on falling edge; i.e., trigger when measurement goes from above the limit to below it or when the digital trigger goes from high to low. When triggering on the rising or falling edge, the input must make the specified transition to trigger. For example, when triggering on the rising edge, if the input starts out high, it must go low and then high again to trigger.

DATA SENT TO INPUT STORAGE

When the measurements are sent to Input Storage, Parameter 6 is used to specify the number of scans made on the channels being measured (the CR10 multiplies the number entered by 1000). The measurements from each channel are stored contiguously. For example: Parameter 1 specifies 4 channels are to be measured, Parameter 6 specifies 250 scans (0.250 entered), and Parameter 10 specifies 1 as the first input location to store data. The measurements from the first channel will be stored in input locations 1-250, those from the second channel in locations 251-500, etc. If insufficient locations are allocated to Input Storage (*A) to accommodate the number of locations called for by Parameter 6 multiplied by Parameter 1 (e.g., 250 x 4 = 1000), an error code, E 60, will be displayed when the program is compiled.

The number of scans determines how many samples will be saved, and hence, when Instruction 23 will be completed and execution will pass on to the next instruction. Measurements before and/or after the trigger can be saved. Parameter 7 determines how many scans that occurred prior to the trigger are saved. For example, if 250 scans are specified, and an offset of 20 is entered for Parameter 7, then the trigger measurement will be stored in location 21. Locations 1-20 will contain the measurements that preceded the trigger and 22-250 will contain the measurements following the trigger. If only 10 measurements are made before the trigger, then they will be stored in locations 11-20 while the value -99999. will be stored in locations 1-10 for which no measurements were made.

DATA SENT TO SERIAL PORT

When the raw A/D data is sent out the serial port, the measurement data is not buffered and hence, only the trigger and subsequent measurements can be sent. The number of measurements is determined by Parameters 1 and 6. Because the total number of measurements are limited only by the storage of the receiving computer, this can be a very large number. Parameter 6 is the number of measurements per channel to send in units of 1000 (e.g., 0.001 represents 1 measurement). If 0 is entered for Parameter 6, the CR10 will continue to send data until the Instruction is aborted by pressing any key on the CR10KD Keyboard Display.

Raw A/D data may be sent to a maximum of 8 SM192 or SM716 Solid State Storage Modules (SM's) at 76.8K baud (Parameter 4, Destination Option C = 3). All SM's connected should be set to Fill and Stop and have consecutive addresses. The Burst data will be sent to the first available (lowest address) Storage Module, followed by the next lowest addressed, and so on. If the data is retrieved from the SM's using SMCOM.COM, a Storage Module communications program contained in the PC208 Datalogger Support Software, the data collection format must be "A as stored 8 bit". This will transfer the raw A/D data to a computer file. The raw A/D data can be converted to ASCII using SPLIT.COM, a

general purpose data reduction program also contained in PC208.

If SPLIT is not available for converting the raw A/D, the following A/D format information is provided for decoding purposes. At the start of the series of measurements, the CR10 makes a self-calibration measurement. The calibration data is sent at the start of the measurement data. The serial data is sent as a series of signed 2 byte integers (most significant byte sent first; i.e., Integer = 256 * byte 1 + byte 2): I1...In. The first integer, I1 is a start of output identifier, FCxx (hex), where the first byte is always FC (never seen in the data), and the second byte is a number less than 100 (decimal, 64 hex), which is the Instruction Location Number of Instruction 23 in the program table. I2 divided by I3 is the multiplier and I4 the offset (to the raw data) determined by the first calibration. I2 is a fixed value determined by the input range selected. I5 through In are the raw measurement data. Thus, the value of the first measurement sent (M1) in millivolts is:

$$M_1 = I_2/I_3 (I_5 - I_4)$$

The measurement data are sent in the order that the measurements are made (i.e., the first measurement for each channel, then the second measurement for each channel, etc.).

> **NOTE**: When the raw serial data option is selected, the calibration values are for conversion to millivolts only. Parameters 11 and 12 are ignored.

SCAN INTERVAL
Instruction 23 has its own scan interval independent of the execution interval of the program table in which it resides. The resolution of the clock timing the execution interval is 813 nanoseconds. This scan interval, entered in Parameter 5 (in milliseconds), is the time between each scan of the specified channels (i.e., if 4 channels are specified in Parameter 1, and the scan interval is 5 ms, then the 4 measurements will be repeated every 5 ms). The minimum time that is allowed per measurement is 1.333 ms. The maximum time that is allowed per measurement is 50 ms. If the scan interval entered does not allow this much time per measurement (e.g., if with 4 reps, an interval less than 5.332 ms is entered), an error code, E 61, will be displayed when the program is compiled. When sending data to the serial port, the rate at which the data can be transferred may limit the scan interval (e.g., at 9600 baud the minimum time per measurement is 2.2 ms).

Burst/Telecommunications Considerations

If a Burst measurement sequence is in progress, raising the datalogger's ring line will abort the Burst sequence. Peripherals which raise the ring line are modems (i.e., RF, Telephone, Short Haul, MD9, SC32A) or the CR10KD.

If the Burst Measurement Instruction is encountered while telecommunications is in progress, the destination of the data determines whether or not the instruction is executed:

Burst data sent to input locations

If a CR10 already in Telecommunications mode executes a Burst instruction specifying that Burst data be sent to input locations, all telecommunication activity will be suspended. After the Burst trigger condition is met and all Burst measurements made, telecommunications activity can resume.

Burst data sent to Serial I/O Port

If the Burst instruction specifies that Burst data be sent to the serial port (i.e., Storage Module), CR10 program execution will pause until the Telecommunication mode is exited. During this pause telecommunications (i.e., view input locations, Monitor Mode with Term, etc.) can continue. No Burst measurements are made while in telecommunications and no Burst data is sent to the serial port. After telecommunications has ended, datalogger program execution will resume as if the Burst instruction were just executed.

> **NOTE:** Instruction 23 can be aborted by pressing any key on the CR10KD Keyboard Display.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions (no. of channels)
02:	2	Range code (13-15)

03:	2	Single-ended or differential channel for first analog measurements
04:	4	Option, 4 digit code ABCD A Trigger 0 - Trigger on 1st analog channel 1 - Digital trigger on Control Port #1 2 Same as 0, but set Digital Control Port #1 high when trigger is met, low when done measuring B Trigger option 0 - Trigger immediately 1 - Trigger if above limit (high) 2 - Trigger if below limit (low) 3 - Trigger on rising edge 4 - Trigger on falling edge C Destination 0 - Input Storage 1 - Serial port 9600 baud 2 - Serial port 76,800 baud 3 - Serial port 76,800 baud to SM192/716 D Measurement 0 - Differential measurement 1 - Single-ended measurement
05:	FP	Scan interval (ms, minimum 1.333 x reps, limited to 1.333-50 ms)
06:	FP	Number of scans (units of 1000)
07:	4	Number of samples saved before trigger (not used with serial output)
08:	FP	Trigger limit (mV, unscaled measurement)
09:	4	Excitation voltage (mV)
10:	4	1st input location in which to store data
11:	FP	Multiplier (not used with serial output)
12:	FP	Offset (not used with serial output)

*** 24 CALIBRATION ***

FUNCTION
Put 19 calibration values into input locations. If C (--) is keyed before entering the input location, then the automatic calibrations are simply displayed, not measured. Otherwise, the calibration takes place only when Instruction 24 is executed; automatic calibration is disabled (Section 13.7).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location number, key C (--) for results of automatic calibration

Input locations altered: 19

*** 25 PORT READ ***

FUNCTION
The status of a group of ports selected by a mask is read and placed in an input location. The status is a base 2 representation of the ports converted to base 10. Port 1 is the least significant bit. For example, if all ports are read, and the port status is as follows:

PORT	C8	C7	C6	C5	C4	C3	C2	C1
VALUE	128	64	32	16	8	4	2	1
STATUS (0=low, 1=high)	0	0	1	1	0	0	1	0

Base 10 equivalent: 32 + 16 + 2 = 50
50 will be stored in the input location.

The mask is also base 2 representation; 1 indicates the port is to be read, 0 results in a 0 for the port regardless of the status of the port (AND operation). For example, if 50 (see above example) is entered for the mask, ports 2, 5, and 6 are read. If only ports 4 and 5 are high, the status will be 16 (port 4 is not read).

NOTE: Voltages in excess of 5.5 volts applied to a control port can cause the CR10 to malfunction.

PARAM. NUM.	DATA TYPE	DESCRIPTION
01:	4	MASK (0-255)
02:	4	INPUT LOCATION TO STORE RESULT

Input locations altered: 1

***** 26 TIMER *****

FUNCTION
This instruction will reset a timer or store the elapsed time registered by the timer in seconds in an Input Storage location. Instruction 26 can be used with Program Control Instructions to measure the elapsed time between specific input conditions. There is only one timer and it is common to all tables (e.g., if the timer is reset in Table 1 and later in Table 2, a subsequent instruction in Table 1 to read the timer will store the elapsed time since the timer was reset in Table 2).

Elapsed time is tracked in 0.125 second increments. The maximum interval that can be timed is 8191.875 seconds.

The timer is also reset in response to certain keyboard entries:

1. When tables are changed and compiled with the *0 Mode, the timer is reset automatically.
2. Entering "*6" after changing the program compiles the programs, but does NOT reset the timer.

PARAM. NUM.	DATA TYPE	DESCRIPTION
01:	4	Input location no. of elapsed time in seconds (or enter 0 to reset)

Input locations altered: 1
(0 if timer is being reset)

***** 27 PERIOD MEASUREMENT *****

FUNCTION
Instruction 27 measures the period (microseconds) of a signal on a single ended input channel. As an option, the frequency of the signal in kHz may be output instead of the period. The specified number of cycles are timed with a resolution of +60 nanoseconds, making the resolution on the period 60 nanoseconds divided by the number of cycles measured. Resolution is reduced by noise and signals with a slow transition through the zero voltage threshold. The "Time out" parameter specifies the maximum length of time the instruction will wait on each repetition for the specified number of cycles. If the cycles have not been counted within this time, -99999 will be loaded into the input location.

Input Frequency Gain Codes

Range Code	Peak to Peak Volts Required @ Max. Freq.*	Maximum Frequency
1	2 mV	8 kHz
2	3 mV	20 kHz
3	12 mV	50 kHz
4	2 V	200 kHz

0x Output period in microseconds
1x Output frequency in kHz where x is range code

* AC voltage; must be centered around CR10 ground.

PARAM. NUM.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Gain/output option
03:	2	Single-ended input channel
04:	4	# Cycles to measure
05:	4	Time out (0.01 sec, at least the maximum duration of the number of cycles specified + 1 1/2 cycles.)
06:	4	Destination input location
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1* Repetitions

***** 28 VIBRATING WIRE *** MEASUREMENT**

FUNCTION
Excites a vibrating wire sensor with a swept frequency (from low frequency to high), then measures the response period and calculates $1/T^2$, where T is the period in ms. Excitation is normally provided for before each repetition. As an option, a single excitation can be made prior to all repetitions

of the measurement. An AVW1 or AVW4 Vibrating Wire Interface is usually required for these sensors.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions Hit C (--) to skip repeat of excitation
02:	2	Single-ended channel for first measurement
03:	2	Excitation Channel
04:	2	Start frequency of sweep (100'S of Hz)
05:	2	End frequency of sweep (100'S of Hz)
06:	4	# Cycles to measure (0 means none)
07:	4	Delay before excitation applied (0.01 sec units)
08:	4	Input location ($1/T^2$), T in ms
09:		FP Multiplier
10:		FP Offset

***** 101 SDM-INT8 *****

FUNCTION
The 8 channel Interval Timer (INT8) is a measurement module which provides processed timing information to the datalogger. Each of the 8 input channels may be independently configured to detect either rising or falling edges of either a low level AC signal of a 5 V logic signal. Each channel may be independently programmed. See the SDM-INT8 manual for detailed instructions and examples.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Address of INT8
02:	4	*Input config; channels 8,7,6,5
03:	4	*Input config; channels 4,3,2,1
04:	4	**Function; channels 8,7,6,5
05:	4	**Function; channels 4,3,2,1
06:	4	***Averaging option
07:	4	Loc
08:	FP	Mult
09:	FP	Offset

Execution Time: 2.3 ms + 1.65 ms/value + averaging interval if used (See Appendix C for possible extra processing time for higher frequency signals.)

Intermediate Storage: 1 location

* Input configurations (Sect. 4.2):

0 =	5V logic level, rising edge
1 =	5V logic level, falling edge
2 =	low level ac, rising edge
3 =	low level ac, falling edge

** Function (Sect. 4.3):

0 =	none
1 =	period in ms
2 =	frequency in Khz
3 =	time since previous channel's edge in ms
4 =	time since channel 1 in ms
5 =	counts on channel 2 since channel 1, linear interpolation
6 =	frequency in Khz (low resolution)
7 =	counts
8 =	counts on Channel 2 since Channel 1, no interpolation

*** Averaging option (Sect. 4.4):

0	Average over execution interval
0-	Continuous averaging
XXXX	Averaging interval in ms, XXXX>0
XXXX-	Capture all events until XXXX edges of channel 1 (0<XXXX, 9999)
9999-	Test memory

***** 102 SDM-SW8A *****

FUNCTION
The 8 channel SDM-SW8A Switch Closure Input Module is a peripheral for measuring up to 8 channels of switch closure or voltage pulse inputs. Each channel may be configured to read single-pole double-throw (SPDT) switch closure, single-pole single-throw (SPST) switch closure, or voltage pulse. Output options include counts, duty cycle, or state.

The SW8A is addressed by the datalogger, allowing multiple SW8A's to be connected to one datalogger. 16 addresses are available, but for most applications Campbell Scientific recommends no more than 4 SW8A's be connected to one datalogger.

If more channels are requested than exist in one module, the datalogger automatically increments the address and continues to the next SW8A. The address settings for multiple SW8A's must sequentially increase. For example, assume 2 SW8A's with an address of 22 and 23 are connected, and 12 Reps are requested. Eight channels from the first SW8A and the first 4 channels from the second SW8A will be read.

Only one Function Option (Parameter 3) may be specified per P102 Instruction. If all 4 functions are desired, 4 P102 Instructions are required.

Function Option 0 provides the state of the signal at the time P102 is executed. A 1 or 0 corresponds to high or low states, respectively.

Function Option 1 provides signal duty cycle. The result is the percentage of time the signal is high during the sample interval.

Function Option 2 provides a count of the number of positive transitions of the signal.

Function Option 3 provides the signature of the SW8A PROM. A positive number (signature) indicates the PROM and RAM are good, a zero (0) indicates bad PROM, and a negative number indicates bad RAM. Function Option 3 is not used routinely, but is helpful in "debugging". Only one Rep is required for Option 3.

Parameter 4 specifies the first SW8A channel to be read (1..8). One or more sequential channels are read depending on the Reps. To optimize program efficiency, the sensors should be wired sequentially.

Data are stored in sequential input locations, starting at the location specified in Parameter 5. The number of input locations consumed is equal to the number of Reps.

The scaling multiplier and offset (Parameters 6 and 7) are applied to all readings. If a multiplier is not entered, all readings are set to 0.

If the SW8A does not respond, -99999 will be loaded into input locations. Modules which do not respond when addressed by the datalogger are possibly wired or addressed incorrectly. Verify the address specified in Parameter 2 corresponds to the jumper setting and all connections are correct and secure. See the SDM-SW8 Manual for examples.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Number of Channels
02:	2	Module Address (00..33)
03:	2	Function Option (0=State, 1=Duty, 2=Counts, 3=Signature)
04:	2	SDM-SW8A Starting Channel (1..8)
05:	4	Starting input location
06:	FP	Mult
07:	FP	Offset

***** 103 SDM-AO4 *****

FUNCTION
Instruction 103 is used to activate a SDM-AO4 connected to ports C1, C2, and C3.

There are 4 analog voltage outputs per SDM-AO4. The output voltages in millivolts must be stored in 4 adjacent input locations starting with the location entered in parameter 3. Four reps are required for each SDM-A04. Every 4 reps another device at the next higher address is selected.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01	2	Reps (# of outputs)
02	2	Address (Base 4)
03	4	Starting input location

C1 is Data line
C2 is Clk/Hand Shake line
C3 is SDE (Enable) line

***** 104 SDM-CD16AC *****

FUNCTION
The SDM-CD16AC Control Port Expansion Module has 16 digital control ports with drivers. Each port can be controlled by a datalogger or controlled manually with an override toggle switch. Each port can be thought of as a switch to ground; closed when active, open when inactive. The primary function is to activate AC powered external relays, solenoids, or resistive loads under datalogger control.

The SDM-CD16AC is a synchronously addressed datalogger peripheral. Datalogger control ports 1, 2 and 3 are used to address the SDM-CD16AC then clock out the desired state

of each of the 16 control ports. Up to 16 SDM-CD16AC's may be addressed, making it possible to control a maximum of 256 ports from the first three datalogger control ports.

For each Rep, the 16 ports of the addressed SDM-CD16AC are sent according to 16 sequential input locations starting at the input location specified in parameter 3. Any non-zero value stored in an input location activates (connects to ground) the associated SDM-CD16AC port. A value of zero (0) deactivates the port (open circuit). For example, assuming 2 Reps and a starting input location of 33, OUTPUT 1 through 16 of the first SDM-CD16AC are set according to input locations 33 through 48, and OUTPUT 1 through 16 of the second SDM-CD16AC are set according to input locations 49 through 64. See the SDM-CD16AC manual for detailed instructions and examples.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Reps (# of CD16AC modules sequentially addressed)
02:	2	Starting Address (base 4: 00..33)
03:	4	Starting input location

***** 105 SDI-12 RECORDER *****

Instruction 105 allows data to be collected from a SDI-12 sensor. A single Control Port is connected to the SDI-12 sensor data line; the SDI-12 ground should be connected to a "G" terminal on the CR10 wiring panel. The CR10 may also be used as the 12 volt power supply for sensors. If multiple SDI-12 sensors are used, each sensor requires a separate Instruction 105. Instruction 105 is only in standard PROMs OS10-1.1 and OS10-1.2.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	ADDRESS (0-9)
02:	2	COMMAND (Table 1)
03:	2	PORT (1-8)
04:	4	INPUT LOC.
05:	FP	MULT
06:	FP	OFFSET

Input locations altered: 0-9 (SDI-12 sensor dependent)

Intermediate Locations required: 9

PARAMETER 1. ADDRESS

Enter the address of the SDI-12 sensor (0-9). Non-standard addresses may be used entering the appropriate ASCII code.

PARAMETER 2. COMMAND

Enter a number to select the command to be sent to the SDI-12 sensor. Usually 0 is entered to select the M command. The following Standard SDI-12 commands are supported by the CR10:

TABLE 9-6. SDI-12 Command Codes

ENTRY	COMMAND	DESCRIPTION
O	M	Initiate Measurement
1..9	M1..M9	Additional Measurement commands specified by the SDI-12 sensor
10	V	Initiate Verify sequence
11	I	Send Identification

The result of a M, M1-M9, or V command sequence is numerical data, stored in an input location. The response to the I command is text information, which is written directly to Final Storage regardless of the Output Flag's state. The *7 mode of the datalogger cannot be used with this type of data.

PARAMETER 3. PORT

Enter the CR10 control port connected to the SDI-12 sensor data line. Any port 1-8 (C1-C8) may be used. Default port is C8.

PARAMETER 4. INPUT LOCATION

Input location where the returned data is stored. If multiple values are returned from the SDI-12 sensor they are stored in sequential input locations beginning at the specified location.

ERRORS

If the CR10 receives either an incorrect response or no response is received from a SDI-12 sensor, the CR10 will retry the operation. If, after the retries, a valid response has not been received, the CR10 will store a -99999 in the input location specified in Parameter 4. Only the first location will be

altered. Sequential locations will contain values from previous measurements.

TRANSPARENT MODE

The SDI-12 transparent mode is used to communicate directly with a SDI-12 sensor. A common application of the transparent mode is to verify proper SDI-12 sensor operation.

A computer or terminal is required to use the transparent mode; the CR10KD (keyboard display) cannot be used. Transparent mode is entered while the computer is in telecommunications with the SDI-12 recorder CR10 (at the asterisk '*' prompt). Enter 'pX' at the asterisk prompt, where 'p' is the Control Port number (1-8) attached to the SDI-12 data line. The CR10 responds with a '<' prompt. Any SDI-12 command preceded with the sensor address and followed with an exclamation point '!' may then be entered. For example, entering '0I!' at the '<' prompt would request identification from a SDI-12 sensor addressed at 0.

The '<' prompt will not appear until the recorder CR10 finishes executing all program tables. While in transparent mode, scheduled tables in the recorder CR10 will not execute. Transparent mode ends when a response is received and displayed, if no response is received from the SDI-12 sensor within the time-out period following a valid command (approximately 1/3 second), or if the user does not enter a command before the mode times out (approximately 35 seconds). Security must be unlocked to level 2 before the Transparent mode is used.

***** 106 SDI-12 SENSOR *****

Instruction 106 allows a CR10 to be used as a SDI-12 sensor. The CR10 can make measurements and transfer data using SDI-12 commands in response to another SDI-12 recorder. Instruction 106 is only in standard PROM OSI0-1.1.

Instruction 106 supports the standard SDI-12 commands as listed in the Parameter 2 description for Instruction 105.

The SDI-12 data line is attached to Control Port 8 and Instruction 106 must be the first instruction in Subroutine 98 located in Table 3.

A SDI-12 recorder addresses the SDI-12 sensor CR10 by sending a Break and the sensor's address. The sensor CR10 will call subroutine 98 whenever it detects activity on the SDI-12 data line attached to Port 8, but if the Break and the specified address are not received by Instruction 106, the remainder of the subroutine is not executed.

Two programming techniques exist for obtaining measurement values to be transferred by the sensor Instruction 106. The first technique makes the requested measurements "on demand" in response to the recorders request. The measurement instructions are located in Subroutine 98 and are executed only when the SDI-12 recorder requests measurements. This technique is preferred when measurements are to be made at the recorder's command.

The second technique transfers measurement values previously obtained by instructions in Table 1 or Table 2. Subroutine 98 contains only Instructions 106 (SDI-12 sensor) and 95 (End). When the recorder requests measurements, values already in the specified input locations are used. The advantage of this technique is that the sensor CR10 can be making and storing measurements independent of the SDI-12 recorder. The data is also returned slightly faster since the sensor CR10 does not make measurements when the recorder requests data, but rather uses measurements made at the last regular table execution.

These two techniques can be combined allowing the sensor CR10 to function as a SDI-12 sensor and to make independent measurements. While Subroutine 98 is being executed, normal Table 1 or 2 execution scheduling may be altered or missed since Subroutine 98 is not interrupted. This is likely to occur if Subroutine 98 execution takes longer than the scan interval programmed for Table 1 or 2. It is also possible for instructions in Table 1 or 2 to prevent Subroutine 98 from being called in time for Instruction 106 to receive the address information from the recorder. This is likely to occur only if Table 1 or 2 is executed often and has instructions that take longer than 1/3 second to execute. For example, Instruction 4 (Excite-Delay-SE) with a 1/2 second delay could cause Subroutine 98 to miss the SDI-12 address information if it were executing when the SDI-12

data line became active. If this occurs the sensor CR10 will not respond to the SDI-12 recorder. Most instructions execute fast enough that when Instruction 106 misses the initial SDI-12 address, a subsequent retry by the recorder will work.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	ADDRESS (0-9)
02:	4	TIME/VALUES tttn: ttt=time(sec) n=no. values
03:	4	LOCATION starting loc. for n values

Input locations altered: 0

Intermediate locations required: 82. This requires that the default setting of 64 be increased. The *A mode (second window) is used to specify Intermediate locations.

PARAMETER 1. ADDRESS

Enter the address for the CR10 acting as a SDI-12 sensor (0-9). Each SDI-12 sensor must have a unique address. Non standard addresses may be used by entering the appropriate ASCII code.

PARAMETER 2. TIME/VALUES

Enter the time required for the sensor CR10 to complete subroutine 98 followed by the number of input locations to be returned to the SDI-12 recorder. The format of the entry must be 'tttn' where ttt specifies the time and n is the number of input locations values.

The time to complete subroutine 98 is the time required to execute all instructions from Instruction 106 (SDI-12 sensor) to the final Instruction 95 (End). The entered time may be longer than the actual time without slowing the data exchange because the sensor CR10 signals the SDI-12 data recorder when the data is ready for transfer. If the entered time is too short the sensor CR10 will not respond properly and the data values will not be transferred. Enter a time of 0 to transfer the values already stored in input locations, in this case the remainder of the subroutine is not executed.

A maximum of 9 input locations can be transferred at one time since 'n' is a single digit. See Parameter 3 LOCATION description for information on transferring more than 9 locations.

PARAMETER 3. LOCATION

This parameter determines the starting input location for the 'n' values to be returned to the recorder. The 'M' or 'M1-M9' command issued by the SDI-12 recorder determines if the starting location is actually that specified in Parameter 3 or a multiple of 'n' past Parameter 3.

Starting input location = Parameter 3 + (n*x), where n is specified in Parameter 2, and, x is the number following the 'M' sent by the SDI-12 recorder (1-9).If the 'M' command is sent by the recorder x = 0.

Results of Instruction 106

The sensor CR10 will return a set of input locations in response to the M or M1..M9 command sequence. The set of Locations returned is determined by Parameters 2 and 3 of Instruction 106.

The three values, sent in response to a V command sequence, indicate the status of the sensor CR10. The first and second values are from the *B mode of the sensor CR10, giving the number of watchdog errors (E08) and the number of table overruns that have occurred. The third is a signature of the sensor CR10 memory. This signature is created by the same technique that the Instruction 19 (Signature) uses. The CR10 manual contains more information on *B and Instruction 19.

In response to an I command, the sensor CR10 sends the string '10CAMPBELL CR10 001 aaaa' where aaaa is the number from the eighth window of the *B mode.

SECTION 10. PROCESSING INSTRUCTIONS

To facilitate cross referencing, parameter descriptions are keyed [ ] to the values given on the PROMPT SHEET. These values are defined as follows:

[Z] = Destination input location for result
[X] = Input location of X
[Y] = Input location of Y
[F] = Fixed Data (user specified floating point number)

*** 30 LOAD FIXED DATA ***

FUNCTION
Store a fixed value into an input location. The value is entered in scientific notation; the absolute value of the number may range from 1 x 10^{-19} to 9 x 10^{18}. A value smaller than the minimum is set to 0, while a larger value is set to the maximum.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	FP	Mantissa	[F]
02:	2	Exponent of 10 (Hit C to change sign)	
03:	4	Destination for input location	[Z]

Input locations altered: 1

*** 31 COPY INPUT DATA ***

FUNCTION
Copy data from one input location to another.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Source input location number	[X]
02:	4	Destination input location	[Z]

Input locations altered: 1

*** 32 INCREMENT INPUT LOCATION ***

FUNCTION
Add 1 to the current value in the specified input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Destination input location	[Z]

Input locations altered: 1

*** 33 X + Y ***

FUNCTION
Add X to Y and place result in a third input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Input location of Y	[Y]
03:	4	Dest. input location of X+Y	[Z]

Input locations altered: 1

*** 34 X + F ***

FUNCTION
Add F to X (where F is a fixed floating point number) and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	FP	Fixed value	[F]
03:	4	Dest. input location of X+F	[Z]

Input locations altered: 1

*** 35 X - Y ***

FUNCTION
Subtract Y from X and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Input location of Y	[Y]
03:	4	Dest. input location for X-Y	[Z]

Input locations altered: 1

*** 36 X * Y ***

FUNCTION
Multiply X by Y and place the result in an input location (Z).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Input location of Y [Y]
03:	4	Dest. input location for X*Y [Z]

Input locations altered: 1

*** 37 X * F ***

FUNCTION
Multiply X by F (where F is a fixed multiplier) and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	FP	Fixed value [F]
03:	4	Dest. input location for X*F [Z]

Input locations altered: 1

*** 38 X / Y ***

FUNCTION
Divide X by Y and place the result in an input location. Division by 0 will cause the result to be set to the maximum CR10 number (99999).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Input location of Y [Y]
03:	4	Dest. input location for X/Y [Z]

Input locations altered: 1

*** 39 SQUARE ROOT ***

FUNCTION
Take the square root of X and place the result in an input location. If X is negative, 0 will be stored as the result.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input location for $X^{1/2}$ [Z]

Input locations altered: 1

*** 40 LN(X) ***

FUNCTION
Take the natural logarithm of X and place the result in an input location. If X is 0 or negative, -99999 will be stored as the result.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input location for LN(X) [Z]

Input locations altered: 1

*** 41 EXP(X) ***

FUNCTION
Raise the exponential (EXP) base e to the X power and place it in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input for EXP(X) [Z]

Input locations altered: 1

*** 42 1/X ***

FUNCTION
Take the inverse of X and place the result in an input location. If X=0, 99999 will be given as the result.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input location for 1/X [Z]

Input locations altered: 1

*** 43 ABS(X) ***

FUNCTION
Take the absolute (ABS) value of X and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input location for ABS(X) [Z]

Input locations altered: 1

*** 44 FRACTIONAL VALUE ***

FUNCTION
Take the fractional (FRAC) value (i.e., the non-integer portion) of X and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input location for FRAC(X) [Z]

Input locations altered: 1

*** 45 INTEGER VALUE ***

FUNCTION
Take the integer (INT) value of X and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input location for INT(X) [Z]

Input locations altered: 1

*** 46 X MOD F ***

FUNCTION
Do a modulo divide of X by F and place the result in an input location. X MOD F is defined as the REMAINDER obtained when X is divided by F (e.g., 3 MOD 2 = 1). X MOD 0 returns X.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	FP	Fixed divisor [F]
03:	4	Dest. input loc. For X MOD F [Z]

Input locations altered: 1

*** 47 X^Y ***

FUNCTION
Raise X to the Y power and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Input location of Y [Y]
03:	4	Dest. input location for X^Y [Z]

Input locations altered: 1

*** 48 SIN(X) ***

FUNCTION
Calculate the sine of X (X is assumed to be in degrees) and place the result in an input location. The cosine of a number can be obtained by adding 90 to the number and taking the sine (COSX = SIN (X + 90)).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input location for SIN(X) [Z]

Input locations altered: 1

*** 49 SPATIAL MAXIMUM ***

FUNCTION
Find the spatial maximum (SPA MAX) value of the given set or SWATH of input locations and place the result in an input location. To find the input location where the maximum value occurs, add 1000 to the input location number destination selected [Z] and enter this modified location number as Parameter 03. The input location ID of the maximum value observed will then be stored in destination [Z] plus 1.

Parameter 3 cannot be entered as an indexed location within a loop (Instruction 87). To use Instruction 49 within a loop, enter Parameter 3 as a fixed location and follow 49 with the Instruction 31 (Move Data). In Instruction 31, enter the location in which 49 stores its result as the source (fixed) and enter the destination as an indexed location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Swath [SWATH]
02:	4	Starting input location [1ST LOC]
03:	4	Dest. input location for maximum [MAX or Z]

Input locations altered: 1 or 2

***** 50 SPATIAL MINIMUM *****

FUNCTION
Find the spatial minimum (SPA MIN) value of the given set or SWATH of input locations and place the result in an input location. To find the input location where the minimum value occurs, follow the instructions given above for SPATIAL MAXIMUM.

Parameter 3 cannot be entered as an indexed location in a loop. Within a loop, Instruction 50 must be used in conjunction with Instruction 31 as described for Instruction 49.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Swath [SWATH]
02:	4	Starting input location [1ST LOC]
03:	4	Dest. input location for minimum [MIN or Z]

Input locations altered: 1 or 2

***** 51 SPATIAL AVERAGE *****

FUNCTION
Take the spatial average (SPA AVG) over the given set or SWATH of input locations and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Swath [SWATH]
02:	4	Starting input location [1ST LOC]
03:	4	Dest. input location of average [AVG or Z]

Input locations altered: 1

***** 53 SCALING ARRAY WITH *** MULTIPLIER AND OFFSET**

FUNCTION
Take 4 input location values, multiply each by a floating point constant, then add another floating point constant to the resulting products and place the final results back into each of the original 4 input locations.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	First input location	[STRT LOC]
02:	FP	Multiplier 1	[A1]
03:	FP	Offset 1	[B1]
04:	FP	Multiplier 2	[A2]
05:	FP	Offset 2	[B2]
06:	FP	Multiplier 3	[A3]
07:	FP	Offset 3	[B3]
08:	FP	Multiplier 4	[A4]
09:	FP	Offset 4	[B4]

Input locations altered: 4

***** 54 BLOCK MOVE *****

FUNCTION
Executes a "block move" of data in input locations. Parameters specify the number of values to move, the source, source step, destination, and destination step. The "step" parameters designate the increment of the source and destination input locations for each value that is moved. For example, a "source step" of 2 and a "destination step" of 1 will move data from every other input location to a contiguous block of input locations.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Number of values to move
02:	4	1st source location
03:	2	Step of source
04:	4	1st destination location
05:	2	Step of destination

Intermediate storage: 0

*** 55 5TH ORDER POLYNOMIAL ***

FUNCTION
Evaluate a 5th order polynomial of the form

$$F(X)=C0+C1X+C2X^2+C3X^3+C4X^4+C5X^5$$

where C0 through C5 are the coefficients for the argument X raised to the zero through fifth power, respectively. The magnitude of the user entered coefficient is limited to a range of ±.00001 to ±99999. Polynomials with coefficients outside this range can be modified by pre-scaling the X value by an appropriate factor to place the coefficients within the entry range. Pre-scaling can also be used to modify coefficients which are very close to 0 to increase the number of significant digits.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions [REPS]
02:	4	Starting input location for X [X]
03:	4	Dest. input location for F(X) [F(X) or Z]
04:	FP	C0 coefficient [C0]
05:	FP	C1 coefficient [C1]
06:	FP	C2 coefficient [C2]
07:	FP	C3 coefficient [C3]
08:	FP	C4 coefficient [C4]
09:	FP	C5 coefficient [C5]

Input locations altered: 1* Reps

*** 56 SATURATION VAPOR PRESSURE ***

FUNCTION
Calculate saturation vapor pressure (over water SVPW) in kilopascals from the air temperature (°C) and place it in an input location. The algorithm for obtaining SVPW from air temperature (°C) is taken from: Lowe, Paul R.: 1977, "An approximating polynomial for computation of saturation vapor pressure," *J. Appl. Meteor,* **16**, 100-103.

Saturation vapor pressure over ice (SVPI) in kilopascals for a 0°C to -50°C range can be obtained using Instruction 55 and the relationship

$$SVPI = -.00486 + .85471\ X + .2441\ X^2$$

where X is the SVPW derived by Instruction 56. This relationship was derived by Campbell Scientific from the equations for the SVPW and the SVPI given in Lowe's paper.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of air temperature °C [TEMP.]
02:	4	Destination input location for saturated vapor pressure [VP or Z]

Input locations altered: 1

*** 57 VAPOR PRESSURE FROM WET-/DRY-BULB TEMPERATURES ***

FUNCTION
Calculate vapor pressure in kilopascals from wet and dry-bulb temperatures in °C. This algorithm type is used by the National Weather Service:

$$VP = VPW - A(1 + B*TW)(TA - TW)\ P$$

VP = ambient vapor pressure in kilopascals
VPW = saturation vapor pressure at the wet-bulb temperature in kilopascals
TW = wet-bulb temperature, °C
TA = ambient air temperature, °C
P = air pressure in kilopascals
A = 0.000660
B = 0.00115

Although the algorithm requires an air pressure entry, the daily fluctuations are small enough that for most applications a fixed entry of the standard pressure at the site elevation will suffice. If a pressure sensor is employed, the current pressure can be used.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of atmospheric pressure in kilopascals [PRESSURE]
02:	4	Input location of dry-bulb temp. [DB TEMP.]
03:	4	Input location of wet-bulb temp. [WB TEMP.]
04:	4	Destination input location for vapor pressure [VP or Z]

Input locations altered: 1

***** 58 LOW PASS FILTER *****

FUNCTION
Apply a numerical approximation to an analog resistor capacitor (RC) low pass (LP) filter using the following algorithm.

$$F(Xi) = W*Xi + F(Xi-1) * (1-W)$$

Where X = input sample, W = user entered weighting function (O< W <1). If W=O, $F(Xi)=X_1$; if W=1, F(Xi)=X and F(Xi-1) = output calculated for previous sample

The equivalent RC time constant is given by T/W, where T is the sampling time in seconds. For values of W less than 0.25, the analogous "cut off" frequency (the frequency where the ratio of output to input is .707) is accurately represented by $W/(2\pi T)$. For larger values of W, this "analog" estimate of the cutoff frequency becomes less representative.

On the first execution after compiling, F(X) is set equal to X.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions [REPS]
02:	4	First input location for input data [X]
03:	4	Dest. input location for first filtered result[F(X) or Z]
04:	FP	Weighting function, W [W]

Input locations altered: 1 for each repetition

***** 59 BRIDGE TRANSFORM *****

FUNCTION
This instruction is used to aid in the conversion of a ratiometric Bridge measurement by obtaining the value for R_s which is equivalent to $R_f[X/(1-X)]$, where X is the value derived by the standard CR10 Bridge Measurement Instructions (with appropriate multiplier and offset, Section 13.5) and R_f represents the MULTIPLIER value. The result of Instruction 59 is stored in the same location that X was.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions [REPS]
02:	4	Starting input location & result destination [X]
03:	FP	Multiplier (Rf) [MULT]

Input locations altered: 1 for each repetition

***** 61 INDIRECT INDEXED MOVE *****

FUNCTION
Moves input data from location X to location Y, where X and/or Y are indirectly addressed (X and Y are stored in the locations specified by Parameters 1 and 2). If a location parameter is specified as "indexed" (xxxx--), then the actual input location referenced is calculated by adding the current index counter to the value in the specified input location. When used outside a loop, the addressing is simply indirect because the index counter is zero.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Source input location
02:	4	Dest. input location

Input locations altered: 1

***** 63 PARAMETER EXTENSION *****

Instruction 63 is used immediately following Instructions 97 or 98 to allow the entry of a variable number of parameters. Instruction 63 can be entered several times in sequence if the number of parameters requires it. There are 8 two digit parameters. Refer to instruction being extended (97, 98) for specifics on the use of Instruction 63.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:-08:	2	Depends upon preceding instruction. Following Instruction 97 RF IDs and Phone No. 1 digit at a time, 32 Between RF Ids, 72 Between RF and DC112 Phone, and No. 13 To END. Following Instruction 98 (255 character limit) Base 10 value of ASCII character (Appendix E) 00 TO END.

Input locations altered: 0

***** 64 PAROSCIENTIFIC "T" SERIES PROCESSING *****

FUNCTION
Instruction 64 processes measurements made on the Paroscientific "T" Series pressure transducer. Instruction 105 is in standard PROM OS10-1.1 and OS10-2.1. The transducer outputs a temperature frequency and a pressure frequency. The frequencies are measured using the Period Averaging Instruction, 27, which returns the period of the temperature and pressure signals in microseconds. Instruction 64 converts period to temperature and pressure according to the following relationships provided by Paroscientific.

Pressure (psi) =
$C * (1 - T_0^2/Tau^2) * (1 - (D * (1 - T_0^2/Tau^2)))$

Temperature (°C) =

$Y_1U + Y_2U^2 + Y_3U^3$

where, $C = C_1 + C_2U + C_3U^2$ (psi),

$T_0 = T_1 + T_2U + T_3U^2 + T_4U^3 + T_5U^4$ (microsecond),

$D = D_1 + D_2U$ (microsecond),

$U = U(t) - U_0$ (microsecond),

Tau = measured pressure (microsecond),
U(t) = measured temperature (microsecond).

Values for the calibration coefficients (U_0, Y_1, Y_2, Y_3, C_1, C_2, C_3, D_1, D_2, T_1, T_2, T_3, T_4, T_5) are provided by Paroscientific.

Instruction 64 has two parameters as shown below.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Start Loc: Temp(us), Pressure(us), U0..T5
02:	4	Start Loc: Temp(°C), Pressure(psi), Signature

PARAMETER 1 - Tau, U(t) and 14 coefficients provided by Paroscientific must reside in contiguous input locations starting at the location entered in Parameter 1.

PARAMETER 2 - The results of Instruction 64 (temperature in °C, pressure in psi, and signature of the 14 coefficients) are stored in 3 contiguous input locations starting at the location entered in Parameter 2.

Only the first 5 digits of each 7 digit coefficient are entered in the datalogger. The maximum error that occurs due to using five digit coefficients is 0.001 psi throughout a 15 psi range at any temperature between -30 and +60°C. The maximum error in temperature calculations is 0.2°C over the same temperature range.

Coefficients are entered using CR10 Instruction 30 (Z=F). A calibration sheet from Paroscientific lists the 14 coefficients. Coefficients are entered in the same order as they appear on the calibration sheet. Coefficient U_0 is entered first and T_5 last. Y_3/1000 and D_1*1000 are entered for the Y3 and D1 coefficients, respectively. The example below shows the correct format for entering the coefficients.

Example: The 14 coefficients shown below are for Paroscientific "T" Series transducer Serial Number 30135. Your coefficients will be different.

Coeff.	Value	Entry
U_0	5.860253	5.8603
Y_1	-3970.348	-3970.3
Y_2	-7114.265	-7114.3
* Y_3	102779.1	102.78
C_1	70.29398	70.294
C_2	6.610141	6.6101
C_3	-119.2867	-119.29
* D_1	0.0308837	30.884
D_2	0.0	0.0
T_1	26.33703	26.337
T_2	0.8516985	0.85170
T_3	21.80118	21.801
T_4	0.0	0.0
T_5	0.0	0.0

* Y_3 and D_1 coefficients are entered as Y_3/1000 and D_1*1000.

The large number of coefficients which must be entered into the program often leads to entry error. As a check on coefficient entry, the third value output by Instruction 64 is the signature of the coefficients. The signature is a calculated number unique to the value and order of the 14 coefficients. For future comparisons, note this signature after entering the coefficients correctly. Signatures will be identical if and only if exactly the same coefficients are entered.

*** 65 BULK LOAD ***

FUNCTION
Instruction 65 inputs given values in up to eight Input Storage locations. Instruction 65 is only in standard PROM OS10-1.1.

The Bulk Load instruction has 9 parameters. The first eight are the values to be entered in input storage locations. The ninth is the input location for the first data value; subsequent data values are placed in sequential input locations.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	FP	Param. to be entered
02:	FP	"
03:	FP	"
04:	FP	"
05:	FP	"
06:	FP	"
07:	FP	"
08:	FP	"
09:	4	Starting input location

*** 66 ARCTAN ***

FUNCTION
Calculate the angle in degrees whose tangent is X/Y. The polarity of X and Y must be known to determine the quadrant of the angle, as shown here. If 0 is entered for Parameter 2, the Arctangent of X is the result (limits of ARCTAN(X) are -90° < ARCTAN < 90°).

Quadrant	Sign of X	Sign of Y
I	+	+
II	+	-
III		-
IV		+

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Input location of Y [Y]
03:	4	Destination input location for ARCTAN(X/Y)

Input locations altered: 1

SECTION 11. OUTPUT PROCESSING INSTRUCTIONS

*** 69 WIND VECTOR ***

FUNCTION
Instruction 69 processes the primary variables of wind speed and direction from either polar (wind speed and direction) or orthogonal (fixed East and North propellers) sensors. It uses the raw data to generate the mean wind speed, the mean wind vector magnitude, and the mean wind vector direction over an output interval. Two different calculations of wind vector direction (and standard deviation of wind vector direction) are available, one of which is weighted for wind speed.

When used with polar sensors, the instruction does a modulo divide by 360 on wind direction, which allows the wind direction (in degrees) to be 0 to 360, 0 to 540, less than 0, or greater than 540. The ability to handle a negative reading is useful in an example where a difficult to reach wind vane is improperly oriented and outputs 0 degrees at a true reading of 340 degrees. The simplest solution is to enter an offset of -20 in the instruction measuring the wind vane, which results in 0 to 360 degrees following the modulo divide.

When a wind speed sample is 0, the instruction uses 0 to process scalar or resultant vector wind speed and standard deviation, but the sample is not used in the computation of wind direction. The user may not want a sample less than the sensor threshold used in the standard deviation. If this is the case instruction 89 can be used to check wind speed, and if less than the threshold, Instruction 30 can set the input location equal to 0.

Standard deviation can be processed one of two ways: 1) using every sample taken during the output period (enter 0 for parameter 2), or, 2) by averaging standard deviations processed from shorter sub-intervals of the output period. Averaging sub-interval standard deviations minimizes the effects of meander under light wind conditions, and it provides more complete information for periods of transition[1].

Standard deviation of horizontal wind fluctuations from sub-intervals is calculated as follows:

$$\sigma(\Theta)=[((\sigma\Theta_1)^2+(\sigma\Theta_2)^2 \ldots+(\sigma\Theta_M)^2)/M]^{1/2}$$

where $\sigma(\Theta)$ is the standard deviation over the output interval, and $\sigma\Theta_1 \ldots \sigma\Theta_M$ are sub-interval standard deviations.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Samples per sub-interval (number of scans, enter 0 for no sub-interval)
03:	2	Sensor/Output 2 digits: AB A Sensor type: 0 = Speed and Direction 1 = East and North B Output option: 0 S, Θ1, σ(Θ1) 1 S, Θ1 2 S, U, Θu, σ(Θu)
04:	4	First wind speed input location no. (East wind speed)
05:	4	First wind direction input location no. (North wind speed)

Outputs Generated: 2-4 (depending on output option) for each repetition

A sub-interval is specified as a number of scans. The number of scans for a sub-interval is given by:

Desired sub-interval (secs) / scan rate (secs)

---

[1] EPA On-site Meteorological Program Guidance for Regulatory Modeling Applications.

In an example where the scan rate is 1 second and the Output Flag is set every 60 minutes, the standard deviation is calculated from all 3600 scans when the sub-interval is 0. With a sub-interval of 900 scans (15 minutes) the standard deviation is the average of the four sub-interval standard deviations. The last sub-interval is weighted if it does not contain the specified number of scans.

There are three Output Options that specify the values calculated.

Option 0:

Mean horizontal wind speed, S.
Unit vector mean wind direction, $\Theta 1$.
Standard deviation of wind direction, $\sigma(\Theta 1)$.

Standard deviation is calculated using the Yamartino algorithm. This option complies with EPA guidelines for use with straight-line Gaussian dispersion models to model plume transport.

Option 1:

Mean horizontal wind speed, S**.**
Unit vector mean wind direction, $\Theta 1$.

Option 2:

Mean horizontal wind speed, S.
Resultant mean wind speed, U.
Resultant mean wind direction, Θu.
Standard deviation of wind direction, $\sigma(\Theta u)$.

This standard deviation is calculated using Campbell Scientific's wind speed weighted algorithm.

Use of the Resultant mean horizontal wind direction is not recommended for straight-line Gaussian dispersion models, but may be used to model transport direction in a variable-trajectory model.

Measured raw data:

S_i = horizontal wind speed
Θ_i = horizontal wind direction
Ue_i = east-west component of wind
Un_i = north-south component of wind
N = number of samples

Calculations:

Scalar mean horizontal wind speed, S:

$$S=(\Sigma S_i)/N$$

where in the case of orthogonal sensors:

$$S_i=(Ue_i^2+Un_i^2)^{1/2}$$

Unit vector mean wind direction, $\Theta 1$:

$$\Theta 1=\text{Arctan } (Ux/Uy)$$

where

$$Ux=(\Sigma \sin \Theta_i)/N$$

$$Uy=(\Sigma \cos \Theta_i)/N$$

or, in the case of orthogonal sensors

$$Ux=(\Sigma(Ue_i/U_i))/N$$

$$Uy=(\Sigma(Un_i/U_i))/N$$

$$\text{where } U_i=(Ue_i^2+Un_i^2)^{1/2}$$

Standard deviation of wind direction, σ(Θ1), using Yamartino algorithm:

$$\sigma(\Theta 1)=\text{arc sin}(\varepsilon)[1+0.1547\ \varepsilon^3]$$

where,

$$\varepsilon=[1-((Ux)^2+(Uy)^2)]^{1/2}$$

and Ux and Uy are as defined above.

Resultant mean horizontal wind speed, U:

$$U=(Ue^2+Un^2)^{1/2}$$

where for polar sensors:

$$Ue=(\Sigma S_i \text{ Sin } \Theta_i)/N$$

$$Un=(\Sigma S_i \text{ Cos } \Theta_i)/N$$

or, in the case of orthogonal sensors:

$$Ue=(\Sigma Ue_i)/N$$

$$Un=(\Sigma Un_i)/N$$

Resultant mean wind direction, Θu:

$$\Theta u=\text{Arctan } (Ue/Un)$$

Standard deviation of wind direction, σ(Θu), using Campbell Scientific algorithm:

$$\sigma(\Theta u)=81(1-U/S)^{1/2}$$

*** 70 SAMPLE ***

FUNCTION
This instruction stores the value from each specified input location. The value(s) stored are those in the input location(s) when Instruction 70 is executed with the Output Flag set high.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Repetitions
02:	4	Starting input location no.

Outputs Generated: 1 for each sample

*** 71 AVERAGE ***

FUNCTION
This instruction stores the average value over the given output interval for each input location specified.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Starting input location no.

Outputs Generated: 1 for each input location

*** 72 TOTALIZE ***

FUNCTION
This instruction stores totalized value over the given output interval for each input location specified.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Starting input location no.

Outputs Generated: 1 for each input location

*** 73 MAXIMIZE ***

FUNCTION
This instruction stores the MAXIMUM value taken (for each input location specified) over a given output interval. An internal FLAG is set whenever a new maximum value is seen. This FLAG may be tested by Instruction 79. Time of occurrence maximum value(s) is OPTIONAL output information, which is formatted and activated by entering one of the following CODES for Param. 2.

Code	Options
00	Output value ONLY
01	Output value with SECONDS
10	Output value with HOUR-MINUTE
11	Output value with HR-MIN,SEC

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Time of maximum (optional)
03:	4	Starting input location no.

Outputs Generated: 1 for each input location (plus 1 or 2 with time of max. option)

*** 74 MINIMIZE ***

FUNCTION
Operating in the same manner as Program 73, this instruction is used for storing the MINIMUM value (for each input location specified) over a given output interval.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Time of minimum (optional)
03:	4	Starting input location no.

Outputs Generated: 1 for each input location (plus 1 or 2 with time of min. option)

*** 75 STANDARD AND WEIGHTED *** VALUE HISTOGRAM

FUNCTION
Processes input data as either a standard histogram (frequency distribution) or a weighted value histogram.

The standard histogram outputs the fraction of the Output Interval that the value in a specified input location (defined as the bin select value) is within a particular sub-range of the total specified range. A counter in the bin associated with each sub-range is incremented whenever the value falls within that sub-range. The value which is output to Final Storage for each bin is computed by dividing the accumulated total in each bin by the total number of scans. This form of output is also referred to as a frequency distribution.

The weighted value histogram uses data from 2 input locations. One location contains the bin select value; the other contains the weighted value. Each time the instruction is executed, the weighted value is added to a bin. The sub-range that the bin select value is in determines the bin to which the weighted value is added. When the Output Flag is set, the value accumulated in each bin is divided by the TOTAL number of input scans to obtain the values that are output to Final Storage. These

values are the contributions of the sub-ranges to the overall weighted value. To obtain the average of the weighted values that occurred while the bin select value was within a particular sub-range, the value output to Final Storage must be divided by the fraction of time that the bin select value was within that particular sub-range (i.e., a standard histogram of the bin select value must also be output).

For either histogram, the user must specify: 1) the number of repetitions, 2) the number of bins, 3) a form code specifying whether a closed or open form histogram is desired (see below), 4) the bin select value input location, 5) the weighted value input location (see below), 6) the lower range limit, and 7) the upper range limit.

The standard histogram (frequency distribution) is specified by entering "0" in the weighted value input location parameter. Otherwise, this parameter specifies the input location of the weighted value. When more than one repetition is called for, the bin select value location will be incremented each repetition and the weighted value location will remain the same (same weighted value sorted on the basis of different bin select values). The weighted value location will be incremented if it is entered as an indexed location (key "C" at some point while keying in Parameter 5; two dashes, --, will appear on the right of the display). At the user's option, the histogram may be either closed or open. The open form includes all values below the lower range limit in the first bin and all values above the upper range limit in the last bin. The closed form excludes any values falling outside of the histogram range.

The difference between the closed and open form is shown in the following example for temperature values:

Lower range limit	10°C
Upper range limit	30°C
Number of bins	10

	Closed Form	Open Form
Range of first bin	10 to 11.99 deg.	<12 deg.
Range of last bin	28 to 29.99 deg.	>28 deg.

A common use of a closed form weighted value histogram is the wind speed rose. Wind speed values (the weighted value input) are accumulated into corresponding direction sectors (bin select input).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Number of bins
03:	2	Form code (0=open form, 1=closed form)
04:	4	Bin select value input location no.
05:	4	Weighted value input location no. (0 = frequency distribution option)
06:	FP	Lower limit of range
07:	FP	Upper limit of range

Outputs Generated: Number of Bins * Repetitions

*** 77 RECORD REAL TIME ***

FUNCTION
This Instruction stores the current time in Final Storage. At midnight the clock rolls over from 23:59 to 00:00. The day also changes.

If hourly or daily summary data is output, it may be desirable to have the previous day output, since that is when the measurements were made. Entering a 2 for the day code causes the previous day to be output if it is the first minute of the day. Similarly, entering 2 for the hour-minute code causes 2400 instead of 0000 to be output (the next minute is still 0001). When day and hour-minute are both output, a 2 for either code results in the previous day at 2400.

The year is output as 19xx if xx is greater than 85, otherwise it will be output as 20xx. The CR10 will require a PROM update in the year 2085. If year is output along with a 2 option in day or hour-minute, the previous year will be output during the first minute of the new year.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Enter appropriate TIME option code

Outputs Generated: 1 for each time parameter selected

Code	Result
xxx1	SECONDS (with resolution of 0.125 sec.)
xx1x	HOUR-MINUTE
xx2x	HOUR-MINUTE, 2400 instead of 0000
x1xx	JULIAN DAY
x2xx	JULIAN DAY, previous day during first minute of new day
1xxx	YEAR

Any combination of Year, Day, HR-MIN, and seconds is possible (e.g., 1011: YEAR, HR-MIN, SEC).

*** 78 SET HIGH OR LOW RESOLUTION *** DATA STORAGE FORMAT

FUNCTION
This instruction enables high resolution (5 character) or low resolution (4 character) final data storage format. Instruction 78 should be entered ahead of the output instructions for which the specified resolution is desired. The default format is low resolution. At the beginning of each program table execution, the low resolution format is automatically enabled.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	0 = low resolution; 1 = high resolution

Outputs Generated: 0

*** 79 SAMPLE ON MAXIMUM OR MINIMUM ***

FUNCTION

Instruction 79 samples specified input location values at the time a new maximum or minimum value is detected by a previous Maximize (73) or Minimize (74) Instruction. When the Output Flag is set, the values copied to Intermediate Storage are transferred to Final Storage.

Instruction 79 must directly follow the maximum or minimum Instruction to which it refers. If the previous Instruction 73 or 74 has more than 1 repetition, Instruction 79 samples whenever a new maximum or minimum is detected in any of the locations. If sampling is to occur only when a specific input location shows a new maximum or minimum, the previous Maximize or Minimize Instruction should have one rep referring to that input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions (number of sequential locations to sample)
02: no.	4	Starting input location

Outputs Generated: 1 for each repetition

*** 80 SET ACTIVE STORAGE AREA ***

FUNCTION

Instruction 80 is used to redirect Output data to either of the Final Storage areas or to Input Storage and to set the array ID for Final Storage. At the beginning of each table the Active Output area is set to Final Storage Area 1.

When directed to Final Storage, the second parameter can be used to set the output array ID. Instruction 80 should follow the instruction setting flag 0. If parameter 2 is 0, the array ID is determined by the instruction location number of Instruction 80 or by the instruction that set the Output Flag, whichever comes last. When data are sent to Input Storage, no array ID is sent.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Storage area option 00 or 01 = Final Storage Area 1 02 = Final Storage Area 2 03 = Input Storage Area
02:	4	Starting input location destination if option 03 Output Array ID if options 0-2 (1-511 are valid IDs)

*** 82 STANDARD DEVIATION IN TIME ***

FUNCTION
Calculate the standard deviation (STD DEV) of a given input location. The standard deviation is calculated using the formula:

$$S = ((\Sigma X_i^2 - (\Sigma X_i)^2/N)/N)^{1/2}$$

where X_i is the ith measurement and N is the number of samples.

PARAM. NUMBER	DATA TYPE	DESCRIPTION

01:	2	Repetitions
02:	4	Starting input location no.

Outputs Generated: 1 for each repetition

SECTION 12. PROGRAM CONTROL INSTRUCTIONS

TABLE 12-1. Flag Description

Flag 0	Output Flag
Flag 1 to 8	User Flags
Flag 9	Intermediate Processing Disable Flag

TABLE 12-2. Command Codes

0	Go to end of program table[3]
1-9, 79-98	Call Subroutine 1-9, 79-99[1]
10-19	Set Flag 0-9 high
20-29	Set Flag 0-9 low
30	Then Do
31	Exit loop if true
32	Exit loop if false
41-48	Set Port 1-8 high[2]
51-58	Set Port 1-8 low[2]
61-68	Toggle Port 1-8[2]
71-78	Pulse Port 1-8[2]

1 97 and 98 are special subroutines which can be called by Control ports 7 and 8 going high; see Instruction 85 for details.

2 The ports can be indexed to the loop counter (Section 8.4).

3 If this command is executed while in a subroutine, execution jumps directly to the end of the table that called the subroutine.

*** 83 IF CASE X < F ***

FUNCTION
If the value in the location specified in the Begin Case Instruction 93 is less than the fixed value entered as parameter 1 then execute the command in parameter 2 then go to the end of the case statement when the next Instruction 83 occurs. Else, continue to next instruction. See Instruction 93 for an example.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	FP	Fixed value
02:	2	Command

*** 85 LABEL SUBROUTINE ***

FUNCTION
This instruction marks the start of a subroutine. Subroutines are a series of instructions beginning with Instruction 85 and terminated with Instruction 95, END. All subroutines must be placed in Table 3 (Subroutine Table). When a subroutine is called by a command in a Program Control Instruction, the subroutine is executed, then program flow continues with the instruction following that which called the subroutine.

Subroutines may be called from within other subroutines (nested). The maximum nesting level for subroutines is 7 deep. Attempts to nest more than 7 deep will not be detected at compilation, but will result in a run time error. When the seventh subroutine attempts to call the eighth, error 31 will be displayed. Execution will not branch to the eighth subroutine; it will continue with the Instruction following that calling the subroutine.

98, 97 PORT INTERRUPT SUBROUTINES. If subroutine 98 (97) is included in Table 3 then Port 8 (7) will cause an interrupt on the rising edge and the subroutine will be executed. These subroutines can also be called from any table.

These subroutines can interrupt Table 1 or 2 or can occur when neither Table is being executed. When the port activating 97 or 98 goes high during the execution of a table, the instruction being executed is completed before the subroutine is run (i.e. it is as if the subroutine was called by the next instruction).

The priority is 98, 97, Table 1, Table 2. If both 97 and 98 are pending (Ports go high at the same time or both go high during the execution of the same instruction in one of the tables), 98 will be executed first. If 97 or 98 has not interrupted a table then neither table can interrupt it. 97 and 98 cannot interrupt each other. However, when 97 or 98 interrupts a table, it is as if the subroutine were in the table (e.g., if 98 interrupts Table 2, either Table 1 or 97 can interrupt it).

While 97 or 98 is being executed as a result of the respective port going high, that port interrupt is disabled (i.e., the subroutine must be completed before the port going high will have any effect).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Subroutine number (1-9, 79-99)

***** 86 DO *****

FUNCTION
This Instruction unconditionally executes the specified command.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Command (Table 12-2)

***** 87 LOOP *****

FUNCTION
Instructions included between the Loop Instruction and the End Instruction (95) are repeated the number of times specified by the iteration count (Parameter 2), or until an Exit Loop command (31,32) is executed by a Program Control Instruction within the Loop. If 0 is entered for the count, the loop is repeated until an Exit Loop command is executed.

The first parameter, delay, controls how frequently passes through the loop are made. Its units are multiples of the table execution interval. A delay of 0 means that there is no delay between passes through the loop. Each time the table is executed all iterations of the loop will be completed and execution will pass on to the following instructions. If the delay is 5, every fifth time that the execution interval comes up, one pass through the loop is made; only those instructions in the loop will be executed and other portions of the table are not executed in the interim. When a loop with delay is executed, the next execution starts at the loop, skipping over any previous and following instructions in the table.

When a fixed number of iterations are executed, the time spent in the loop is equal to the product of the execution interval, delay, and the number of iterations. For example, a loop with a delay of 1 and a count of 5 will take 5 seconds if the execution interval is 1 second. When the loop is first entered, one pass through the loop is made, then the CR10 delays until the next execution interval and makes the second pass through the loop. After making the fifth pass through the loop, there is the fifth delay, after which execution passes to the instruction following the END instruction which goes with the loop.

While in a loop with delay, the table will not be initiated at each execution interval. (However, the overrun decimals will not be displayed.) Some consequences of this are: The Output Flag will not be automatically cleared between passes through the loop. Because Table 2 cannot interrupt Table 1, Table 2 will not be executed while Table 1 is in a loop with delay. Table 1 will not interrupt Table 2 in the middle of an output array. Thus, if the Output Flag is set in Table 2 prior to entering the loop or within the loop, the flag must be specifically cleared before the end of the pass if Table 1 is to be executed.

Input locations for Processing Instructions within a loop can be entered as Indexed locations. An Indexed location causes the input location to be incremented by 1 with each pass through the loop. (The Index counter is added to the location number in the program table.) Input locations which are not indexed will remain constant.

To specify an Indexed location, depress the C key at some point while keying in the digits for the input location and before entering the location with the A key. Two dashes, --, appear in the two right most characters of the display, indicating the entry is Indexed.

When the same output processing is required on values in sequential input locations, it must be accomplished by using the repetitions parameter of the Output Instruction, not by indexing the input location within a loop.

An Output Instruction within a loop is allotted the same number of Intermediate Storage locations as it would receive if it were not in the loop. For example, the average instruction with a single repetition is allotted only two Intermediate locations: one for the number of samples and one for the running total. Each time through the loop the sample counter is incremented and the value in the referenced input location is added to the total. If the input location is indexed, the values from all input locations are added to the same total.

Note that if the Output Flag is set prior to entering the loop in the above example, 10 values will be output. The first will be the average of all the readings in locations 1-10 since the previous output. Because the Intermediate locations are zeroed each time an output occurs, the next nine values will be the current values (samples at the time of output) of locations 2-10.

Loops can be nested. Indexed locations within nested loops are indexed to the inner most loop that they are within. The maximum nesting level in the CR10 is 9 deep. This applies to If Then/Else comparisons and Loops or any combination thereof. An If Then/Else comparison which uses the Else Instruction (94) counts as being nested 2 deep.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Delay
02:	4	Iteration count

The following example involves the use of the Loop Instruction, without a delay, to perform a block data transformation.

The user wants 1 hour averages of the vapor pressure calculated from the wet- and dry-bulb temperatures of 5 psychrometers. One pressure transducer measurement is also available for use in the vapor pressure calculation.

1. The input locations are assigned as follows:
   a) pressure
      Location 10
   b) dry-bulb temperatures
      Location 11-15
   c) wet-bulb temperatures
      Location 16-20
   d) calculated vapor pressure
      Location 21-25
      (Vapor pressure is written over the wet-bulb temperatures.)
2. The program flow is as follows:
   a) Enter the Loop Instruction (#87) with Delay=0 and iteration count=5.
   b) Calculate the vapor pressure with Instruction 57 using a normal location entry of 10 for atmospheric pressure and Indexed locations of 11, 16 and 16 for the dry-bulb, wet-bulb, and calculated vapor pressure, respectively.

c) End loop with Instruction 95.
d) Use the If Time Instruction (#92) to set the Output Flag every hour.
e) Use the Average Instruction (#71) with 5 repetitions starting at input location 21 to average the vapor pressure over the hour.

The actual keyboard entries for the examples are shown below with the first example Instruction location equal to 10. The Input Instructions to make the pressure and temperature measurements are assumed.

TABLE 12-3. Loop Example: Block Data Transform

10:	P87	Beginning of Loop
01:	0	Delay
02:	5	Loop Count
11:	P57	Wet/Dry Bulb Temp to VP
01:	10	Pressure Loc
02:	11--	Dry Bulb Temp Loc DRY BLB#1
03:	16--	Wet Bulb Temp Loc VP #1
04:	21--	Loc [:VP #1]
12:	P95	End
13:	P92	If time is
01:	0	minutes into a
02:	60	minute interval
03:	10	Set high Flag 0 (output)
14:	P71	Average
01:	5	Reps
02:	21	Loc VP #1

The Loop with a delay may be used so that only those instructions within the Loop are executed while certain conditions are met. As a simple example, suppose it is desired to execute one set of instructions from midnight until 6 AM, another set between 6 AM and 4 PM, and a third set between 4 PM and midnight. Between 6 AM and 4 PM, samples are desired every 10 seconds; the rest of the time one minute between samples is sufficient. The execution interval is set to 10 seconds; when a one minute sample rate is desired, a delay of 6 (6 x 10s = 60s) is used in the loop.

TABLE 12-4. Example: Loop with Delay

```
*          1        Table 1 Programs
  01:     10        Sec. Execution Interval

01:       P87       Beginning of Loop
  01:      6        Delay
  02:      0        Loop Count

11:       P86       Do
  01:      1        Call Subroutine 1

12:       P89       If X<=>F
  01:     25        X Loc DAY
  02:      3        >=
  03:      6        F
  04:     31        Exit Loop if true

13:       P95       End

14:       P87       Beginning of Loop
  01:      1        Delay
  02:      0        Loop Count

27:       P86       Do
  01:      1        Call Subroutine 1

28:       P89       If X<=>F
  01:     25        X Loc DAY
  02:      3        >=
  03:     16        F
  04:     31        Exit Loop if true

29:       P95       End

30:       P87       Beginning of Loop
  01:      6        Delay
  02:      0        Loop Count

36:       P86       Do
  01:      1        Call Subroutine 1

37:       P89       If X<=>F
  01:     25        X Loc DAY
  02:      3        >=
  03:      5        F
  04:     32        Exit Loop if false

38:       P95       End

39:         P       End Table 1

*           3       Table 3 Subroutines

01:       P85       Beginning of Subroutine
  01:      1        Subroutine Number
02:       P18       Time
  01:      2        Hours into current year
                    (maximum 8784)
  02:     24        Mod/by
  03:     25        Loc [:DAY     ]

03:       P95       End
```

*** 88 IF X COMPARED TO Y ***

FUNCTION
This Instruction compares two input locations and, if the result is true, executes the specified Command. The comparison codes are given in Table 12-5.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location for X
02:	2	Comparison code (Table 12-5)
03:	4	Input location for Y
04:	2	Command (Table 3.8-1)

TABLE 12-5. Comparison Codes

Parameter 1	Function
1	IF X = Y
2	IF X ≠ Y
3	IF X ≥ Y
4	IF X < Y

*** 89 IF X COMPARED TO F ***

FUNCTION
This Instruction compares an input location to a fixed value and, if the result is true, performs the specified Command. The comparison codes are given in Table 12-5.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location for X
02:	2	Comparison code (Table 12-5)
03:	FP	Fixed value
04:	2	Command (Table 3.8-1)

*** 90 STEP LOOP INDEX ***

FUNCTION
When used within a Loop (Instruction 87), Instruction 90 will increment the index counter by a specified amount after the first time through the loop, thus affecting all indexed input location parameters in subsequent instructions. For example, if 4 is specified, the index counter will count up by 4 (0,4,8,12,...) inside the loop. Instruction 90 does not affect the loop counter which still counts by 1.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Increment for the loop index counter

***** 91 IF FLAG / PORT *****

FUNCTION
This Instruction checks the status of one of the ten Flags or one of the eight ports and conditionally performs the specified Command.

The first Parameter specifies the condition to check:

1X	Execute command if Flag X is high
2X	Execute command if Flag X is low
4X	Execute command if port X is high
5X	Execute command if port X is low

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Flag or Port condition to check
02:	2	Command (Table 3.8-1)

***** 92 IF TIME *****

FUNCTION
The user specifies the number of minutes or seconds into an interval, the duration of the interval, and a command. The command is executed each time the real time is the specified time into the interval. The "If" condition will always be false if 0000 is entered as the time interval.

The time interval is synchronized with real time; if a 60 minute time interval is specified with 0 minutes into the interval, the Command will be executed each hour on the hour. The time interval is synchronized internally by making a modulo divide (Instruction 46) of the number of minutes since midnight by the specified real time interval. If the result is 0, the interval is up. Thus, the first interval of the day always starts at midnight (0 minutes).

The time into an interval is only true the first time Instruction 92 is executed within a given minute (or second). For example, if the command is to set the Output Flag at 0 minutes into a 10 minute interval, and the execution interval of the table is 10 seconds, every 10 minutes there will only be one output generated by this instruction, not five.

The time into interval and the interval may be entered in seconds for intervals less than 60 seconds.

To enter the times in seconds, press "C" after keying in the number of seconds into the interval for Parameter 1; two dashes will appear to the right of the number (XXXX--). When the time into interval is entered as seconds (XXXX--), the time interval will also be interpreted as seconds.

The Output Flag (Flag 0) is a special case in that it will automatically be cleared if it is not time to set it.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Time into interval (minutes or seconds if entered XXXX--)
02:	4	Time interval (minutes or seconds if time into interval is XXXX--)
03:	2	Command (Table 3.8-1)

***** 93 BEGIN CASE STATEMENT *****

The value in the specified input location is compared against parameters in following If Case instructions (83). When a comparison is true, the command in the If Case instruction is executed and the program flow goes to the End instruction (95) associated with the Begin Case instruction.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location for subsequent comparisons

EXAMPLE:

```
01:    P93        Case
  01:    2        Case Loc

02:    P83        If Case Location < F
  01:   69.4      F
  02:    3        Call Subroutine 3
       else
03:    P83        If Case Location < F
  01:   72        F
  02:   10        Set high Flag 0 (output)
```

```
        else
04:     P83         If Case Location < F
  01:     77.3      F
  02:     30        Then Do
05:     P30         Z=F
  01:     0         F
  02:     0         Exponent of 10
  03:     25        Z Loc :

06:     P95         End   Then Do

07:     P95         End   of Case Statement
```

***** 94 ELSE *****

FUNCTION
When Command 30 (Then/Else) is used with an If Instruction, the Else Instruction is used to mark the start of the instructions to execute if the test condition is false (Figure 3.8-1). The Else Instruction is optional; when it is omitted, a false comparison will result in execution branching directly to the End Instruction. Instruction 94 has no parameters.

***** 95 END *****

FUNCTION
Instruction 95 is used to indicate the end/return of a subroutine (Instruction 85), the end of a loop (Instruction 87), the end of an If Then/Else sequence (Instructions 88-92 when used with command 30), or the end of a Case sequence (Instruction 93). The End Instruction has no parameters.

***** 96 ACTIVATE SERIAL DATA OUTPUT *****

FUNCTION
Instruction 96 is used to activate tape, Storage Module (SM192/SM716 or Card Storage Module), or serial data (printer) output. Normally Instruction 96 is placed in the program table after all Output Instructions have been entered and is executed each time the Table is executed. In this situation any data sent to Final Storage is output as soon as possible. However, by using Program Control Instructions to allow execution of Instruction 96 only at certain times, the user can control when the output device(s) are active. Instruction 96 allows a choice of serial data format and the selection of Addressed or Pin Enabled device for the serial print output.

A single parameter is used to select whether the instruction is to control the tape, "printer", or Storage Module output, and if the printer is selected, the format and baud rate. The Instruction must be entered separately for each device that is to receive output.

If both Final Storage areas are in use, Instruction 96 will send data from the area which is currently active. Final Storage Area 1 is active at the start of each Table. Instruction 80 can be used to change the active area. The Area set by Instruction 80 remains active until changed by another Instruction 80 or the Table ends (at which time Area 1 becomes the active Area 1). Instruction 80 can also direct output to Input Storage, in which case Instruction 96 assumes Final Storage Area 1.

If the CR10 is already communicating on the 9-pin connector when Instruction 96 is executed, the output request is put in a queue and program execution continues. As the 9-pin connector becomes available, each device in the queue will get its turn.

The request is not put in the queue if the same device is already in the queue. The data contained in the queue (and which determine a unique entry) are baud rate (if applicable), and the Final Storage Area. Instruction 98 to send characters also uses this queue.

When an entry reaches the top of the queue, the CR10 sends all data accumulated since the last transfer to the device up to the location of the DSP at the time the device became active (this allows everything in the queue to get a turn even if data is being stored faster than it can be transferred to a particular device).

Tape output code 00 will not activate the tape until at least 512 Final Storage Locations have been stored. Code 09 will cause any data between the TPTR and the DSP (location at time Instruction 96 is executed) to be written to tape. Code 00 is used for most tape applications (Section 4.1).

The "other Final Storage Area" device option (the non-active area) allows a "fast" Final Storage area to be transferred to the main area on some trigger condition so there is some history recorded prior to the trigger condition (Section 8.8).

The source of data is the currently active Final Storage Area set by Instruction 80 (default = 0 or 1).**NOTE:** All memory pointers are positioned 8to the DSP location when the datalogger compiles a program. For this reason, Always retrieve uncollected data before making program changes.

For example, assume the TPTR lags the DSP by less than 512 data points when the datalogger program is altered. On compiling, the TPTR is positioned with the DSP, losing reference to the data that was intended to be transferred to tape. The data is not automatically transferred and appears as a discontinuity in the data file. Until the ring memory wraps around and data overwrite occurs, the data may be recovered using the *8 Mode. This scenario is also true for the SPTR and data intended for a Storage Module.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Option Device 00 = Tape 09 = Tape all data to current DSP 80 = To the other Final Storage Area (New data since the last time) 81 = The other Final Storage Area (The entire active Final Storage Area)

(x) BAUD RATE CODES

0	300 baud
1	1200 baud
2	9600 baud
3	76800 baud

ADDRESSED PRINT DEVICE, x = Baud code
1x = Printable ASCII
2x = Comma Delineated ASCII
3x = Binary Final Storage Format
7N = Storage Module N (N=1-8; Section 4.5.1)
7N = Output File Mark to Storage Module N

PIN-ENABLED PRINT DEVICE, x = Baud code (SDE pulled high)
4x = Printable ASCII
5x = Comma Delineated ASCII
6x = Binary Final Storage Format

*** 97 INITIATE *** TELECOMMUNICATIONS

Instruction 97 is used to have the CR10 initiate telecommunications in response to certain conditions. When the instruction is executed with the Interrupt disable flag set low, the CR10 will make a call and send the ID number specified in Parameter 8 (in ASCII at the specified baud rate). The ID number will be sent every 4 seconds until the CR10 receives a response or the time specified in Parameter 3 expires. The expected response is to have the ID sent back to the CR10, at which time the CR10 will go into the normal telecommunications mode (Section 4) and the time limit on the call will become inactive. In the normal telecommunications mode, the CR10 waits for commands from the device it called. The CR10 will not send any data without first receiving a command to do so. CSI's TELCOM program (part of the PC208 Datalogger Support Software) enables IBM PC/XT/AT/PS-2's or compatibles to automatically answer calls and retrieve data.

When the CR10 receives a correct character, it restarts the 4 second timer used to determine when the ID is sent. There is then 4 seconds in which the CR10 waits to receive the next digit of the ID before it again sends the ID. The CR10 must receive the ID in the correct order without mistakes. If an incorrect character is detected, the CR10 will immediately send the correct ID. If a correct response is not received within the time allotted in Parameter 3, the CR10 will hang up.

When either the RF or Hayes commands (DC112) options are specified, the time limit on the call (without a correct response) specified in Parameter 3 is timed from the start of the instruction and must include the dialing time.

If the correct response is not received, the CR10 will continue to make calls. The CR10 will repeat the calls at the fast interval specified by Parameter 4 for the number of retries specified in Parameter 5, after which the calls will be attempted at the slow interval specified in Parameter 6. The actual delay between retries for both the fast and slow attempts has a random factor built in, which is added as an offset to the delay specified. The random factor prevents calls from different stations from occurring at the same time. This offset will range between 0 and one half of the delay specified. The resolution of the timer for these delays is the execution interval of the table in

which the alarm call is initiated. The randomized retry time is divided by the execution interval to determine how many times Instruction 97 must be executed before it calls again. The Instruction must be executed each time the table is.

Parameter 2 specifies which user flag (1-8) is to be used as the interrupt disable flag. If this flag is set, Instruction 97 will not initiate an alarm call. If the CR10 is in the process of trying to get through with an alarm call, setting the interrupt disable flag will abort further attempts. Instruction 97 sets this flag when an alarm call has received the correct response. Instruction 97 does not clear this flag; the flag will remain set until cleared by the program or external command. When the flag is cleared, Instruction 97 is reinitialized.

The RF path and/or telephone number is entered by following Instruction 97 with one or more entries of Instruction 63. The RF station IDs and phone numbers are entered 1 digit at a time. Decimal equivalents of certain ASCII characters (Appendix E) are used to identify breaks. Separate RF stations with a space (32). Indicate the switch from RF stations to a telephone number (DC112) with a space (32) followed by a "T" (84). Carriage Return (13) is used to end the series of numbers.

If the call is to go through a RF link to a phone, then the RF Modem is specified in Parameter 1. See the PC208 and RF manual for additional interfacing notes.

When the DC112 Modem (Hayes compatible commands) is specified, the following commands are sent to the modem before the phone number: ATV0, ATS7=180, and ATDT. The first command causes the Modem to respond with digits rather than words. The second command causes the modem to wait for the carrier 180 seconds after calling or answering. The third command causes the Modem to dial the number that follows the command in "Touch Tones". Additional commands can be entered as part of the telephone number (e.g., "," for delay or "P" for pulse dialing). The CR10 will not accept the line feed found in some Hayes "compatible" modems.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Modem option and baud rate code. Left digit specifies the modem being used and the right, the baud rate. Modem: 0 - RF; 1 - Short-haul; 2 - DC112 Baud Rate: 0 - 300; 1 - 1200; 2 - 9600; 3 - 76,800
02:	2	Interrupt disable flag
03:	4	Time limit on call, 1 sec. units
04:	4	Delay between fast retries, 1 sec. units
05:	2	No. of retries at fast rate
06:	4	Delay between slow retries, 1 min. units
07:	4	Input location to store no. of tries
08:	4	ID to send

*** 98 SEND CHARACTER ***

Instruction 98 is used with Instruction 63 to send a character or string of characters (up to 15) to the printer. The printer may be either a addressed or pin-enabled (Section 6.2). Instruction 63 must immediately follow 98. The character or characters to send are entered in Instruction 63 as the decimal equivalents (99 is the maximum number allowed) of the 7 bit ASCII character (sent as 8 bits, no parity). For example, to send the ASCII character control R, 18 would be entered. Enter a null (0) to terminate the string. Appendix E contains a listing of the ASCII characters.

If the 9 pin connector is already active when Instruction 98 is executed, the output request is put in a queue (see Instruction 96).

This instruction can be used to send a control character to activate some listing device. The specified character(s) is sent at the time Instruction 98 and 63 are executed.

PARAM. NUMBER	DATA TYPE	DESCRIPTION

01:	2	1x Addressed Print Device 4x Pin-enabled Print Device x is baud rate code

SECTION 13. CR10 MEASUREMENTS

13.1 FAST AND SLOW MEASUREMENT SEQUENCE

The CR10 makes voltage measurements by integrating the input signal for a fixed time and then holding the integrated value for the analog to digital (A/D) conversion. The A/D conversion is made with a 13 bit successive approximation technique which resolves the signal voltage to approximately one part in 7500 of the full scale range on a differential measurement (e.g., 1/7500 x 2.5 V = 333 uV). The resolution of a single-ended measurement is one part in 3750.

Integrating the signal removes noise that could create an error if the signal were instantaneously sampled and held for the A/D conversion. There are two integration times which can be specified for voltage measurement instructions, the slow integration (2.72 ms), or the fast integration (250 us). The slow integration time provides a more noise-free reading than the fast integration time. Integration time is specified in the Range Code of the measurement instruction. Instructions 1 - 14 RANGE codes:

Slow (2.72 ms Integration time)	**Fast (250 us Integration time)**	**60 Hz rejection**	**50 Hz rejection**	**Full Scale range**
1	11	21	31±	2.5 mV
2	12	22	32±	7.5 mV
3	13	23	33±	25 mV
4	14	24	34±	250 mV
5	15	25	35±	2500 mV

One of the most common sources of noise is 60 Hz from AC power lines. Where 60 Hz noise is a problem, range codes 21 - 25 should be used. Two integrations are made spaced 1/2 cycle apart (Figure 13.2-2), which results in the AC noise integrating to 0. Integration time for the 2500 mV range is 1/10 the integration time for the other gain ranges (2.72 ms). For countries with 50 Hz power Range codes 31 - 35 are used for 50 Hz rejection.

There are several situations where the fast integration time is preferred. The fast integration time minimizes time skew between measurements and increases the throughput rate. The current drain on the CR10 batteries is lower when the fast integration time is used. The fast integration time should always be used with the AC half bridge (Instruction 5) when measuring AC resistance or the output of an LVDT. An AC resistive sensor will polarize if a DC voltage is applied, causing erroneous readings and sensor decay. The induced voltage in an LVDT decays with time as current in the primary coil shifts from the inductor to the series resistance; a long integration time would result in most of the integration taking place after the signal had disappeared.

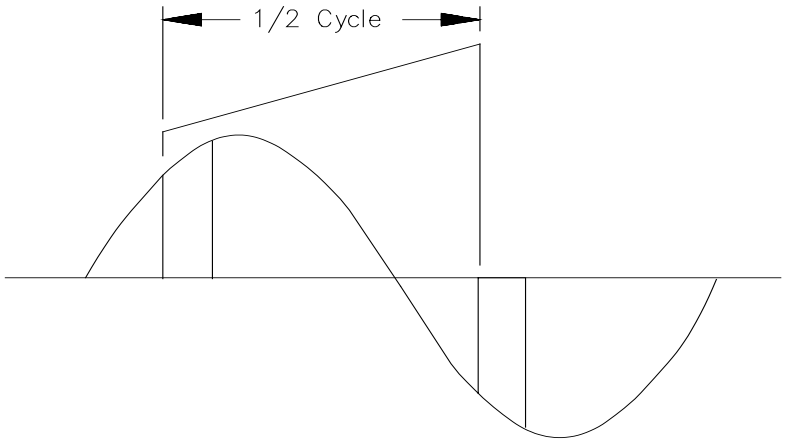


FIGURE 13.1-1. 50 and 60 Hz Noise Rejection

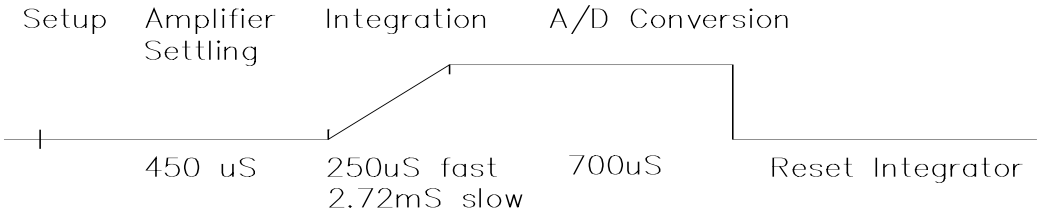


FIGURE 13.2-1. Timing of Single-Ended Measurement

13.2 SINGLE-ENDED AND DIFFERENTIAL VOLTAGE MEASUREMENTS

> **NOTE:** The channel numbering on the old silver CR10 wiring panel refers to differential channels. Either the high or low side of a differential channel can be used for single-ended measurements. Each side must be counted when numbering single-ended channels; e.g., the high and low sides of differential channel 4 are single-ended channels 7 and 8, respectively.

The timing and sequence of a single-ended measurement is shown in Figure 13.2-1. A single-ended measurement is made on a single input which is referenced to ground. A single integration is performed for each measurement. A differential measurement measures the difference in voltage between two inputs. The measurement sequence on a differential measurement involves two integrations. First with the high input referenced to the low, then with the inputs reversed (Figure 13.2-2).

The CR10 computes the differential voltage by averaging the magnitude of the results from the two integrations and using the polarity from the first. An exception to this is the differential measurement in Instruction 8 which makes only one integration.

Because a single-ended measurement is referenced to CR10 ground, any difference in ground potential between the sensor and the CR10 will result in an error in the measurement. For example, if the measuring junction of a copper-constantan thermocouple, used to measure soil temperature, is not insulated and the potential of earth ground is 1 mV greater at the sensor than at the point where the CR10 is grounded, the measured voltage would be 1 mV greater than the thermocouple output or approximately 25°C high.

Another instance where a ground potential difference creates a problem is in a case such as described in Section 7.2, where external signal conditioning circuitry is powered from the same source as the CR10. Despite being tied to the same ground, differences in current drain and lead resistance result in different ground potential at the two instruments. For this reason a differential measurement should be made on an analog output from the external signal conditioner. Differential measurements MUST be used where the inputs are known to be different from ground, such as is the case with the output from a full bridge.

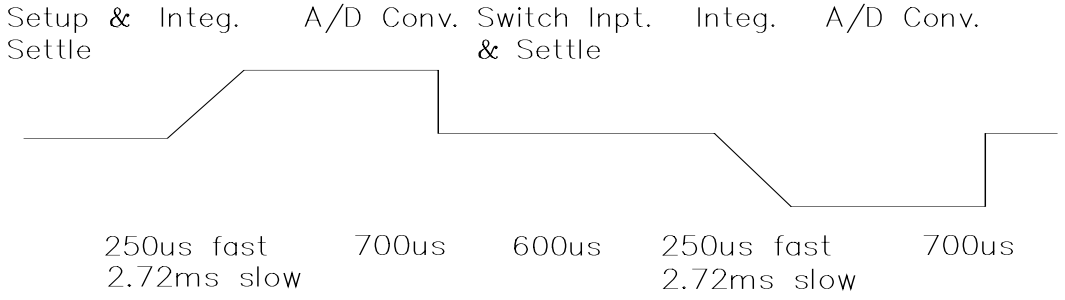


FIGURE 13.2-2. Differential Voltage Measurement Sequence

In order to make a differential measurement, the inputs must be within the CR10 common mode range of ±2.5 V. The common mode range is the voltage range, relative to CR10 ground, within which both inputs of a differential measurement must lie in order for the differential measurement to be made.

For example, if the high side of a differential input is at 2 V and the low side is at 1 V relative to CR10 ground, there is no problem; a measurement made on the +2.5 V range would indicate a signal of 1 V. However, if the high input is at 2.8 V and the low input is at 2 V, the measurement cannot be made because the high input is outside of the common mode range. The CR10 will indicate the overrange with the maximum negative number (Section 3.5.)

Problems with exceeding common mode range may be encountered when the CR10 is used to read the output of external signal conditioning circuitry if a good ground connection does not exist between the external circuitry and the CR10. When operating where AC power is available, it is not always safe to assume that a good ground connection exists through the AC wiring. If a CR10 is used to measure the output from a laboratory instrument (both plugged into AC power and referencing ground to outlet ground), it is best to run a ground wire between the CR10 and the external circuitry. Even with this ground connection, the ground potential of the two instruments may not be at exactly the same level, which is why a differential measurement is desired (Section 7.2).

If a differential measurement is used on a sensor that is not referenced to CR10 ground through a separate connection (e.g., a net radiometer), a jumper wire should be connected between the low side of the differential input and analog ground to hold the sensor in common mode range.

A differential measurement has better noise rejection than a single-ended measurement. Integrating the signal in both directions also reduces input offset voltage due to thermal effects in the amplifier section of the CR10. Input offset voltage on a single-ended measurement is less than 5 microvolts; the input offset voltage on a differential measurement is less than 1 microvolt.

A single-ended measurement is quite satisfactory in cases where noise is not a problem and care is taken to avoid ground potential problems. Channels are available for twice as many single-ended measurements. A single-ended measurement takes about half the time of a differential measurement, which is valuable in cases where rapid sampling is a requirement.

NOTE: Sustained voltages in excess of +16 VDC applied to the analog inputs will damage the CR10 input circuitry.

13.3 THE EFFECT OF SENSOR LEAD LENGTH ON THE SIGNAL SETTLING TIME

Whenever an analog input is switched into the CR10 measurement circuitry prior to making a measurement, a finite amount of time is required for the signal to stabilize at its correct value. The rate at which the signal settles is determined by the input settling time constant which is a function of both the source resistance, and input capacitance (explained below). The CR10 allows a 450 µs settling time before initiating the measurement. In most applications this settling time is adequate, but the additional wire capacitance associated with long sensor leads can increase the settling time constant to the point that measurement errors may occur. There are three potential sources of error which must settle before the measurement is made:

1. The signal must rise to its correct value.
2. A small transient (~5 mV) caused by switching the analog input into the measurement circuitry must settle.
3. A larger transient, usually about 40 mV/V, caused by the switched, precision excitation voltage used in resistive bridge measurements must settle.

The purpose of this section is to bring attention to potential measurement errors caused when the input settling time constant gets too large and to discuss procedures whereby the effects of lead length on the measurement can be estimated. In addition, physical values are given for three types of wire used in CSI sensors, and error estimates for given lead lengths are provided. Finally, techniques are

discussed for minimizing input settling error when long leads are mandatory.

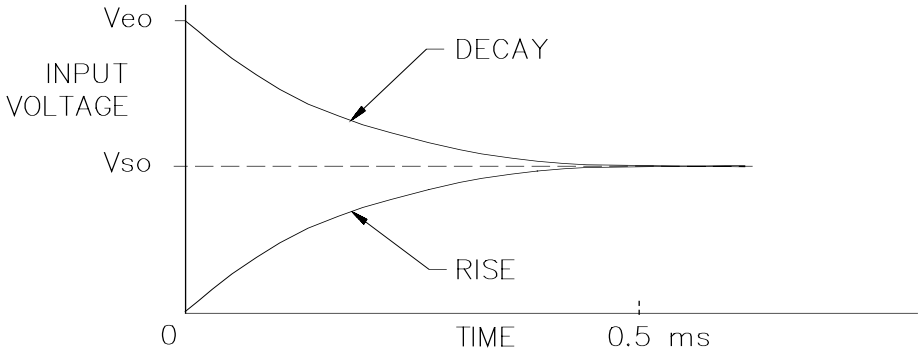


FIGURE 13.3-1. Input Voltage Rise and Transient Decay

13.3.1 THE INPUT SETTLING TIME CONSTANT

The rate at which an input voltage rises to its full value or that a transient decays to the correct input level are both determined by the input settling time constant. In both cases the waveform is an exponential. Figure 13.3-1 shows both a rising and decaying waveform settling to the signal level, Vso. The rising input voltage is described by Equation 13.3-1 and the decaying input voltage by Equation 13.3-2.

$$V_s = V_{so} (1-e^{-t/R_oC_T}), \text{ rise} \qquad [13.3\text{-}1]$$

$$V_s = V_{so} + (V_{eo}-V_{so}) e^{-t/R_oC_T}, \text{ decay} \qquad [13.3\text{-}2]$$

where V_s is the input voltage, V_{so} the true signal voltage, V_{eo} the peak transient voltage, t is time in seconds, R_o the source resistance in ohms, and C_T is the total capacitance between the signal lead and ground (or some other fixed reference value) in farads.

The settling time constant, τ in seconds, and the capacitance relationships are given in Equations 13.3-3 through 13.3-5,

$$\tau = R_oC_T \qquad [13.3\text{-}3]$$

$$C_T = C_f + C_wL \qquad [13.3\text{-}4]$$

$$C_f = 3.3 \text{ nfd} \qquad [13.3\text{-}5]$$

where C_f is the fixed CR10 input capacitance in farads, C_w is the wire capacitance in farads/foot, and L is the wire length in feet.

Equations 13.3-1 and 13.3-2 can be used to estimate the input settling error, V_e, directly. For the rising case, $V_s = V_{so}-V_e$, whereas for the decaying transient, $V_s = V_{so}+V_e$. Substituting these relationships for V_s in Equations 13.3-1 and 13.3-2, respectively, yields expressions in V_e, the input settling error:

$$V_e = V_{so} e^{-t/R_oC_T}, \text{ rise} \qquad [13.3\text{-}6]$$

$$V_e = V_{e'o} e^{-t/R_oC_T}, \text{ decay} \qquad [13.3\text{-}7]$$

Where $V_{e'o} = V_{eo}-V_{so}$, the difference between the peak transient voltage and the true signal voltage.

> **NOTE:** Since the peak transient, V_{eo}, causes significant error only if it is several times larger than the signal, V_{so}, error calculations made in this section approximate $V_{e'o}$ by V_{eo}; i.e., $V_{eo} = V_{eo}-V_{so}$.

If the input settling time constant, τ , is known, a quick estimation of the settling error as a percentage of the maximum error (V_{so} for rising, $V_{e'o}$ for decaying) is obtained by knowing how many time constants (t/τ) are contained in the 450 µs CR10 input settling interval (t). The familiar exponential decay relationship is given in Table 13.3-1 for reference.

TABLE 13.3-1. Exponential Decay, Percent of Maximum Error vs. Time in Units of τ

Time Constants	% Max. Error	Time Constants	% Max. Error
0	100.0	5	0.7
1	36.8	7	0.1
3	5.0	10	0.004

Before proceeding with examples of the effect of long lead lengths on the measurement, a discussion on obtaining the source resistance, R_o, and lead capacitance, C_wL, is necessary.

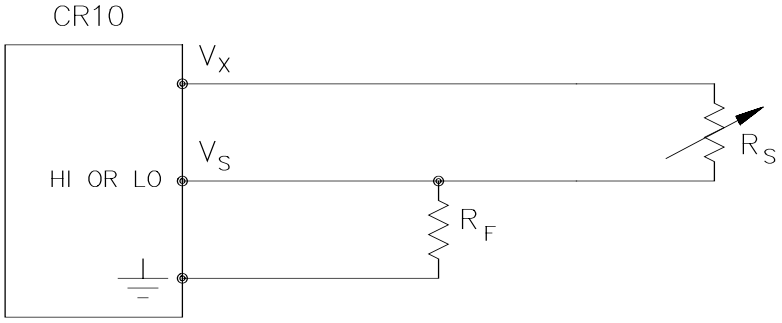


FIGURE 13.3-2. Typical Resistive Half Bridge

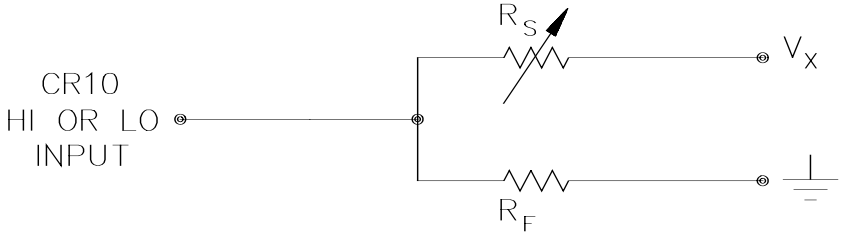


FIGURE 13.3-3. Source Resistance Model for Half Bridge Connected to the CR10

DETERMINING SOURCE RESISTANCE

The source resistance used to estimate the settling time constant is the resistance the CR10 input "sees" looking out at the sensor. For our purposes the source resistance can be defined as the resistance from the CR10 input through all external paths back to the CR10. Figure 13.3-2 shows a typical resistive sensor, (e.g., a thermistor) configured as a half bridge. Figure 13.3-3 shows Figure 13.3-2 re-drawn in terms of the resistive paths determining the source resistance Ro, is given by the parallel resistance of Rs and Rf, as shown in Equation 13.3-8.

$$R_o = R_s R_f/(R_s+R_f) \quad [13.3\text{-}8]$$

If R_f is much smaller, equal to or much greater than R_S, the source resistance can be approximated by Equations 13.3-9 through 13.3-11, respectively.

$$R_o \sim R_f,\ R_f << R_s \quad [13.3\text{-}9]$$

$$R_o = R_f/2,\ R_f = R_s \quad [13.3\text{-}10]$$

$$R_o \sim R_s,\ R_f >> R_s \quad [13.3\text{-}11]$$

The source resistance for several Campbell Scientific sensors are given in column 3 of Table 13.3-5.

DETERMINING LEAD CAPACITANCE

Wire manufacturers typically provide two capacitance specifications: 1) the capacitance between the two leads with the shield floating, and 2) the capacitance between the two leads with the shield tied to one lead. Since the input lead and the shield are tied to ground (often through a bridge resistor, R_f) in single-ended measurements such as Figure 13.3-2, the second specification is used in determining lead capacitance. Figure 13.3-4 is a representation of this capacitance, C_w, usually specified as pfd/ft. C_w is actually the sum of capacitance between the two conductors and the capacitance between the top conductor and the shield. Capacitance for 3 Belden lead wires used in Campbell Scientific sensors is shown in column 6 of Table 13.3-2.

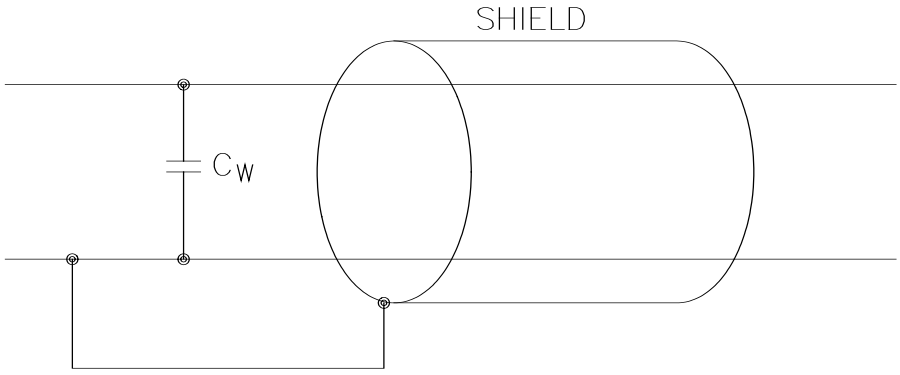


FIGURE 13.3-4. Wire Manufacturers Capacitance Specifications, C_w

TABLE 13.3-2. Properties of Three Belden Lead Wires Used by Campbell Scientific

Belden Wire #	Conductors	Insulation	AWG	Rl (ohms/1000ft.)	C_w (pfd/ft.)
8641	1 shld. pair	polyethylene	24	23	42
8771	1 shld. 3 cond.	polyethylene	22	15	41
8723	2 shld. pair	polypropylene	22	15	62

DIELECTRIC ABSORPTION

The dielectric absorption of insulation surrounding individual conductors can seriously affect the settling waveform by increasing the time required to settle as compared to a simple exponential. Dielectric absorption is difficult to quantify, but it can have a serious effect on low level measurements (i.e., 50 mV or less). The primary rule to follow in minimizing dielectric absorption is: Avoid PVC insulation around conductors. PVC cable jackets are permissible since the jackets don't contribute to the lead capacitance because the jacket is outside the shield. Campbell Scientific uses only polyethylene and polypropylene insulated conductors in CR10 sensors (see Table 13.3-2) since these materials have negligible dielectric absorption. Teflon insulation is also very good but quite expensive.

13.3.2 EFFECT OF LEAD LENGTH ON SIGNAL RISE TIME

In the 024A Wind Vane, a potentiometer sensor, the peak transient voltage is much less than the true signal voltage (see Table 13.3-5). This means the signal rise time is the major source of error and the time constant is the same as if C_w were between the signal lead and ground as represented below.

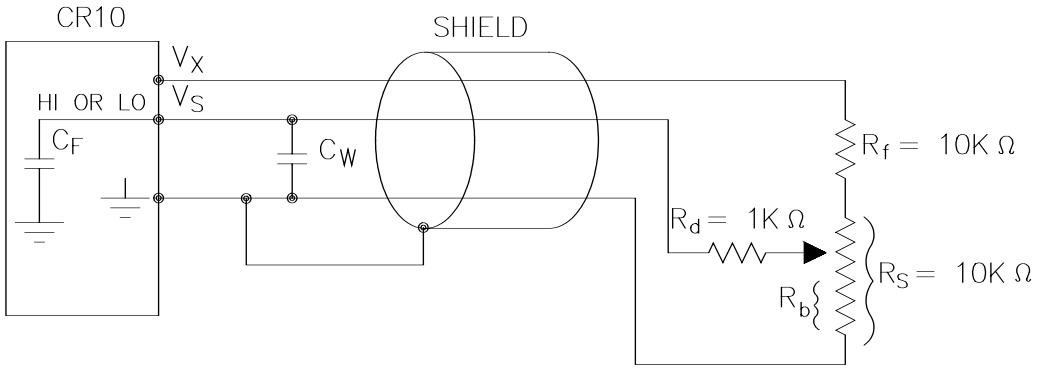


FIGURE 13.3-5. Model 024A Wind Direction Sensor

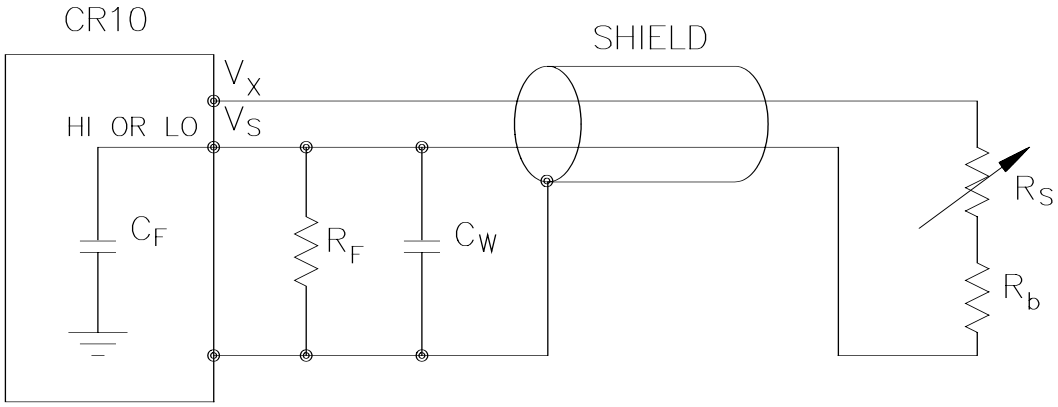


FIGURE 13.3-6. Resistive Half Bridge Connected to Single-Ended CR10 Input

R_o, the source resistance, is not constant because R_b varies from 0 to 10 kohms over the 0 to 360 degree wind direction range. The source resistance is given by:

$$R_o = R_d+(R_b(R_s-R_b+R_f)/(R_s+R_f)) = R_d+(R_b(20k-R_b)/20k) \quad [13.3\text{-}12]$$

Note that at 360 degrees, R_o is at a maximum of 6k (R_b=10k) and at 0 degrees, R_o is 1k (R_b=0). It follows that settling errors are less at lower direction values.

The value of R_b for any direction D (degrees) is given by:

$$R_b(\text{kohms}) = (10k)(D)/360 \quad [13.3\text{-}13]$$

Equation 13.3-6 can be rewritten to yield the settling error of a rising signal directly in units of degrees.

$$\text{Error (degrees)} = De^{-t/(R_o(C_f+C_wL))} \quad [13.3\text{-}14]$$

Equation 13.3-12, -13 and -14 can be combined to estimate the error directly in degrees at various directions and lead lengths (Table 13.3-3). Constants used in the calculations are given below:

C_f = 3.3nfd

C_w = 41 pfd/ft., Belden #8771 wire

t = 450µs

TABLE 13.3-3. Settling Error, in Degrees, for 024A Wind Direction Sensor vs. Lead Length

Wind Direction	Error	
	L=1000 ft.	L=500 ft.
360°	66°	15°
270°	45°	9°
180°	21°	3°
90°	4°	0°

The values in Table 13.3-3 show that significant error occurs at large direction values for leads in excess of 500 feet. Instruction 4, Excite, Delay, and Measure, should be used to eliminate errors in these types of situations. Using a 10 ms delay, settling errors are eliminated up to lengths that exceed the drive capability of the excitation channel (~ 2000 ft.).

13.3.3 TRANSIENTS INDUCED BY SWITCHED EXCITATION

Figure 13.3-6 shows a typical half bridge resistive sensor, such as Campbell Scientific's Model 107 Temperature Probe, connected to the CR10. The lead wire is a single-shielded pair, used for conducting the excitation (V_x) and signal (V_s) voltages. When V_x is switched on, a transient is capacitively induced in V_s, the signal voltage. If the peak transient level, V_{eo}, is less than the true signal, V_{so}, the transient has no effect on the measurement. If V_{eo} is greater than V_{so}, it must settle to the correct signal voltage to avoid errors.

TABLE 13.3-4. Measured Peak Excitation Transients for 1000 Foot Lengths of Three Belden Lead Wires Used by Campbell Scientific

	----------V_{eo}(mV) ----------					
Vx(mV)	**R_f=1 kohm**			**R_f=10 kohm**		
	# **8641**	# **8771**	# **8723**	# **8641**	# **8771**	# **8723**
2000	50	100	60	100	140	80
1000	25	65	40	60	90	40

NOTE: Excitation transients are eliminated if excitation leads are contained in a shield independent from the signal leads.

The size of the peak transient is linearly related to the excitation voltage and increases as the bridge resistor, R_f, increases. Table 13.3-4 shows measured levels of V_{eo} for 1000 foot lengths of three Belden wires used in Campbell Scientific sensors. Values are given for R_f equal to 1 kohm and 10 kohm. Table 13.3-4 is meant only to provide estimates of the size of excitation transients encountered; the exact level will depend upon the specific sensor configuration.

Equation 13.3-7 can be solved for the maximum lead length, L, permitted to maintain a specified error limit. Combining Equations 13.3-7 and 13.3-4 and solving for L gives:

$$L = -(R_o C_f + (t/\ln(V_e/V_{eo})))/R_o C_w \qquad [13.3\text{-}15]$$

where V_e is the measurement error limit.

EXAMPLE LEAD LENGTH CALCULATION FOR 107 TEMPERATURE SENSOR

Assume a limit of 0.05°C over a 0°C to +40°C range is established for the transient settling error. This limit is a reasonable choice since it approximates the linearization error over that range. The output signal from the thermistor bridge varies nonlinearly with temperature ranging from about 100 µV/=°C at 0°C to 50 µV/°C at 40°C. Taking the most conservative figure yields an error limit of V_e = 2.5 µV. The other values needed to calculate the maximum lead length are summarized in Table 13.3-5 and listed below:

1) V_{eo} ~ 50 mV, peak transient at 2 V excitation
2) V_e ~ 2.5 µV, allowable measurement error
3) t = 450 µs, CR10 input settling time
4) R_o = 1 kohm, 107 probe source resistance
5) C_f = 3.3 nfd, CR10 input capacitance
6) C_w ~ 42 pfd/ft., lead wire capacitance

Solving Equation 13.3-15 gives a maximum lead length of:

L ~ 1003 ft., error ~ 0.05°C

Setting the allowable error at 0.1°C or approximately 5 µV, the maximum lead length increases to:

L ~ 1085 ft., error ~ 0.1°C

13.3.4 SUMMARY OF SETTLING ERRORS FOR CAMPBELL SCIENTIFIC RESISTIVE SENSORS

Table 13.3-5 summarizes the data required to estimate the effect of lead length on settling errors for Campbell Scientific's resistive sensors. Comparing the transient level, V_{eo}, to the input range, one suspects that transient errors are the most likely limitation for the 107 sensor. The sensors in the WVU-7 are the same as in the Model 107 (the lead wire is different), but the signal leads for the WVU-7 wet- and dry-bulbs are not subject to excitation transients because they are shielded independently from the excitation.

TABLE 13.3-5. Summary of Input Settling Data For Campbell Scientific Resistive Sensors

Sensor Model #	Belden Wire #	Ro (kohms)	Cw (pfd/ft.)	τ* (us)	Input Range(mV)	V_x(mV)	V_{eo}(mV)**
107	8641	1	42	45	7.5	2000	50
207(RH)	8771	1	41	44	250	1500	85
WVU-7	8723	1	62	65	7.5	2000	0
227	8641	0.1-1	42	5-45	250	250	0
237	8641	1	42	45	25	2500	65
024A	8771	1-6	41	1-222	250	500	0-90

* Estimated time constants are for 1000 foot lead lengths and include 3.3nfd CR10 input capacitance.
** Measured peak transients for 1000 foot lead lengths at corresponding excitation, V_x.

TABLE 13.3-6. Maximum Lead Length vs. Error for Campbell Scientific Resistive Sensors

Sensor Model #	Error	Range	V_e(µV)	Maximum Length(ft.)
107	0.05°C	0°C to 40°C	5	1000[1]
207(RH)	1%RH	20% to 90%	250	2000[3]
WVU-7	0.05°C	0°C to 40°C	5	852[2]
024A	3o	@ 360°	2083	380[2]
227	-	-	-	2000[3]
237	10 kohm	20k to 300k	1000	2000[3]

[1] based on transient settling
[2] based on signal rise time
[3] limit of excitation drive

The comparatively small transient yet large source resistance of the 024A sensor indicates that signal rise time may be the most important limitation. The analysis in Section 13.3.2 confirms this.

The Model 227 Soil Moisture Block has a relatively short time constant and essentially no transient. Lead lengths in excess of 2000 feet produce less than a 0.1 bar (0-10 bar range) input settling error. With this sensor, the drive capability of the excitation channel limits the lead length. If the capacitive load 0.1 µfd and the resistive load is negligible, V_x will oscillate about its control point. If the capacitive load is 0.1 or less, V_x will settle to within 0.1% of its correct value 150 µs. A lead length of 2000 feet is permitted for the Model 227 before approaching the drive limitation.

Table 13.3-6 summarizes maximum lead lengths for corresponding error limits in six Campbell Scientific sensors. Since the first three sensors are nonlinear, the voltage error, V_e, is the most conservative value corresponding to the error over the range shown.

MINIMIZING SETTLING ERRORS IN NON-CAMPBELL SCIENTIFIC SENSORS

When long lead lengths are mandatory in sensors configured by the user, the following general practices can be used to minimize or measure settling errors:

1. When measurement speed is not a prime consideration, Instruction 4, Excite, Delay, and Measure, can be used to insure ample settling time for half bridge, single-ended sensors.

2. An additional low value bridge resistor can be added to decrease the source resistance, R_o. For example, assume a YSI nonlinear thermistor such as the model 44032 is used with a 30 kohm bridge resistor, R_f. A typical configuration is shown in Figure 13.3-7A. The disadvantage with this configuration is the high source resistance shown in column 3 of Table 13.3-7. Adding another 1 K resistor, R_f, as shown in Figure 13.3-7B, lowers the source resistance of the CR10 input. This offers no improvement over configuration A because Rf still combines with the lead capacitance to slow the signal response at point P. The

source resistance at point P (column 5) is essentially the same as the input source resistance of configuration A. Moving R_f' out to the thermistor as shown in Figure 13.3-7C optimizes the signal settling time because it becomes a function of R_f and C_w only.

Columns 4 and 7 list the signal voltages as a function of temperature using a 2000 mV excitation for configurations A and C, respectively. Although configuration A has a higher output signal (2500 mV input range), it does not yield any higher resolution than configuration C which uses the ±250 mV input range.

NOTE: Since R_f' attenuates the signal in configuration B and C, one might consider eliminating it altogether. However, its inclusion "flattens" the non-linearity of the thermistor, allowing more accurate curve fitting over a broader temperature range.

3. Where possible, run excitation leads and signal leads in separate shields to minimize transients.

4. Avoid PVC-insulated conductors to minimize the effect of dielectric absorption on input settling time.

5. Use the CR10 to measure the input settling error associated with a given configuration. For example, assume long leads are required but the lead capacitance, Cw, is unknown. Configure Rf on a length of cable similar to the measurement. Leave the sensor end open as shown in Figure 13.3-8 and measure the result using the same instruction parameters to be used with the sensor. The measured deviation from 0V is the input settling error.

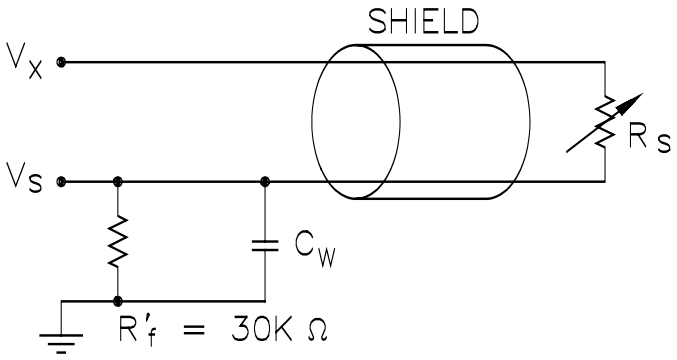
6. Most Campbell Scientific sensors are configured with a small bridge resistor, R_f, (typically 1 kohm) to minimize the source resistance. If the lead length of a Campbell Scientific sensor is extended by connecting to the pigtails directly, the effect of the lead resistance, R_l, on the signal must be considered. Figure 13.3-9 shows a Campbell Scientific Model 107 sensor with 500 feet of extension lead connected directly to the pigtails. Normally the signal voltage is proportional to $R_f/(R_s+R_b+R_f)$, but when the pigtails are extended, the signal is proportional to $(R_f+R_l)/(R_s+R_b+R_f+R_l)$. R_l is much smaller than the other terms in the denominator and can be discarded. The effect on the signal can be analyzed by taking the ratio of the signal with extended leads, V_{sl} to the normal signal, V_s:

$$V_{sl}/V_s = (R_f+R_l)/R_f$$

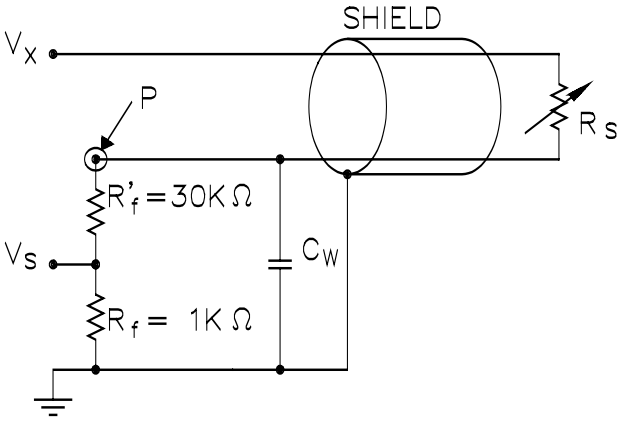
Plugging in values of R_f=1k and R_l=.012k (500' at 23 ohms/1000', Table 13.3-2) gives an approximate 1% error in the signal with extended leads. Converting the error to °C gives approximately a 0.33=°C error at 0°C, 0.53°C error at 20°C, and a 0.66°C error at 40°C. The error can be avoided by maintaining the pigtails on the CR10 end of the extended leads because R_l does not add to the bridge completion resistor, R_f, and its influence on the thermistor resistance is negligible.

TABLE 13.3-7. Source Resistances and Signal Levels for YSI #44032 Thermistor Configurations Shown in Figure 13.3-7 (2V Excitation)

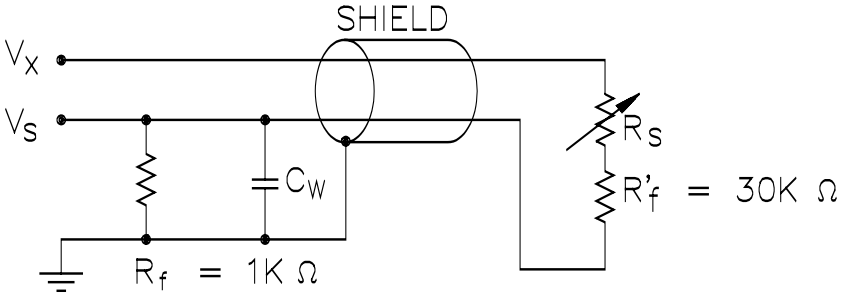
		--------A--------		-----B-----	-------C-------	
T	R_s (kohms)	R_o (kohms)	V_s(mV)	R_o@P (kohms)	R_o (kohms)	V_s(mV)
-40	884.6	29.0	66	30.0	1	2.2
-20	271.2	27	200	27.8	1	6.6
0	94.98	22.8	480	23.4	1	15.9
+25	30.00	15.0	1000	15.2	1	32.8
+40	16.15	10.5	1300	10.6	1	42.4
+60	7.60	6.1	1596	6.1	1	51.8



A) $R_O = R_S R'_f / (R_S + R'_f)$, $V_S = V_X R'_f / (R_S + R'_f)$



B) $R_O @ P = R_S (R'_f + R_f) / (R_S + R'_f + R_f)$



C) $R_O = R_f$, $V_S = V_X R_f / (R_S + R'_f + R_f)$

FIGURE 13.3-7. Half Bridge Configuration for YSI #44032 Thermistor Connected to CR10 Showing: A) large source resistance, B) large source resistance at point P, and C) configuration optimized for input settling

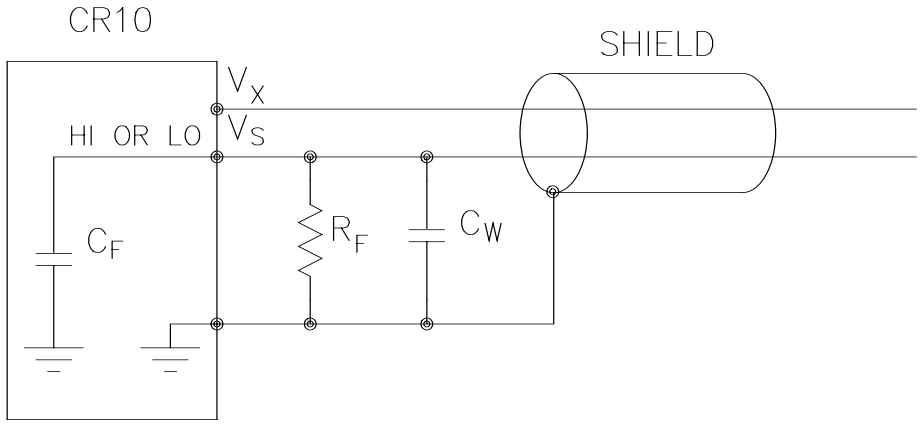


FIGURE 13.3-8. Measuring Input Settling Error with the CR10

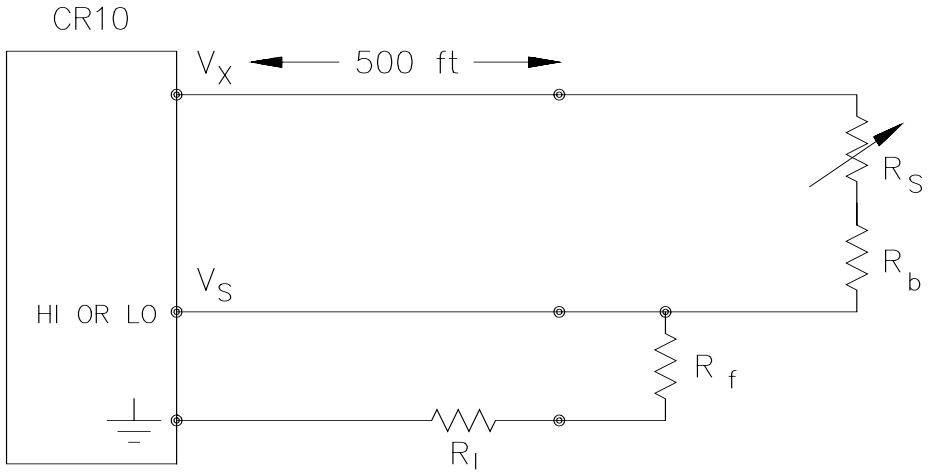


FIGURE 13.3-9. Incorrect Lead Wire Extension on Model 107 Temperature Sensor

13.4 THERMOCOUPLE MEASUREMENTS

A thermocouple consists of two wires, each of a different metal or alloy, which are joined together at each end. If the two junctions are at different temperatures, a voltage proportional to the difference in temperatures is induced in the wires. When a thermocouple is used for temperature measurement, the wires are soldered or welded together at the measuring junction. The second junction, which becomes the reference junction, is formed where the other ends of the wires are connected to the measuring device. (With the connectors at the same temperature, the chemical dissimilarity between the thermocouple wire and the connector does not induce any voltage.) When the temperature of the reference junction is known, the temperature of the measuring junction can be determined by measuring the thermocouple voltage and adding the corresponding temperature difference to the reference temperature.

The CR10 determines thermocouple temperatures using the following sequence. First, the temperature of the reference junction is measured. The reference junction temperature in °C is stored in an input location which is accessed by the thermocouple measurement instruction (Instruction 13 or 14). The CR10 calculates the voltage that a thermocouple of the type specified would output at the reference junction temperature if its reference junction were at 0°C, and adds this voltage to the measured thermocouple voltage. The temperature of the measuring junction is then calculated from a polynomial approximation of the National Bureau of Standards (NBS) TC calibrations.

13.4.1 ERROR ANALYSIS

The error in the measurement of a thermocouple temperature is the sum of the errors in the reference junction temperature, the thermocouple output (deviation from standards published in NBS Monograph 125), the thermocouple voltage measurement, and the polynomial error (difference between NBS standard and CR10 polynomial approximations). The discussion of errors which follows is limited to these errors in calibration and measurement and does not include errors in installation or matching the sensor to the environment being measured.

REFERENCE JUNCTION TEMPERATURE

The junction which is created when a thermocouple is wired to the wiring panel is referred to as the reference junction. The temperature of the reference junction must be known in order to calculate the absolute temperature of the measuring junction. The CR10TCR Thermocouple Reference Temperature (CR10TCR) is used to measure the temperature of the reference junction (terminal strips).

The CR10TCR uses a thermistor to measure the relative temperature of the terminal strips. The accuracy of this measurement is a combination of the thermistor's interchangeability specification, the precision of the bridge resistors, and the polynomial error. In a "worst case" example, all errors add to yield a ±0.4°C error in the range of -24°C to +48°C and ±0.9°C over the range of -38°C to +53°C. It is emphasized that this is the worst case. Campbell Scientific's experience shows that the overall accuracy is typically better than ±0.2°C. The major error component in the -0°C to +50°C range is the ±0.2°C thermistor interchangeability specification. When a CR10 is outside of this temperature range, the polynomial error becomes much worse (Figure 13.4-1), and may necessitate the use of an external reference junction to improve the accuracy.

If the terminal that the thermocouple is wired into is at a different temperature than the CR10TCR thermistor, this difference in temperature becomes an error in the thermocouple temperature measurement. With the CR10 in one of the enclosure options (Section 14) this error will generally be less than 0.3°C.

THERMOCOUPLE LIMITS OF ERROR

The standard reference which lists thermocouple output voltage as a function of temperature (reference junction at 0°C) is the National Bureau of Standards Monograph 125 (1974). The American National Standards Institute has established limits of error on thermocouple wire which is accepted as an industry standard (ANSI MC 96.1, 1975). Table 13.4-1 gives the ANSI limits of error for standard and special grade thermocouple wire of the types accommodated by the CR10.

TABLE 13.4-1. Limits of Error for Thermocouple Wire (Reference Junction at 0°C)

Thermocouple Type	Temperature Range °C	Limits of Error (Whichever is greater) Standard	Special
T	-200 to 0	± 1.0°C or 1.5%	
	0 to 350	± 1.0°C or 0.75%	± 0.5°C or 0.4%
J	0 to 750	± 2.2°C or 0.75%	± 1.1°C or 0.4%
E	-200 to 0	± 1.7°C or 1.0%	
	0 to 900	± 1.7°C or 0.5%	± 1.0°C or 0.4%
K	-200 to 0	± 2.2°C or 2.0%	
	0 to 1250	± 2.2°C or 0.75%	± 1.1°C or 0.4%

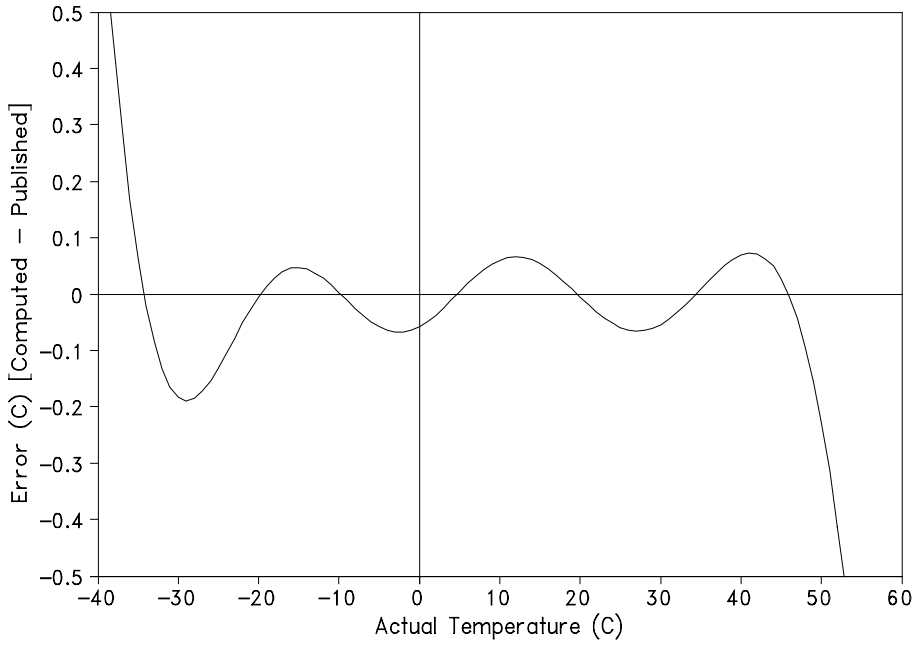


FIGURE 13.4-1. Thermistor Polynomial Error

When both junctions of a thermocouple are at the same temperature, there is no voltage produced (law of intermediate metals). A consequence of this is that a thermocouple cannot have an offset error; any deviation from a standard (assuming the wires are each homogeneous and no secondary junctions exist) is due to a deviation in slope. In light of this, the fixed temperature limits of error (e.g., +1.0°C for type T as opposed to the slope error of 0.75% of the temperature) in the table above are probably greater than one would experience when considering temperatures in the environmental range. In other words, the reference junction, at 0°C, is relatively close to the temperature being measured, so the absolute error (the product of the temperature difference and the slope error) should be closer to the percentage error than the fixed error. Likewise, because thermocouple calibration error is a slope error, accuracy can be increased when the reference junction temperature is close to the measurement temperature. For the same reason, differential temperature measurements, over a small temperature gradient, can be extremely accurate.

In order to quantitatively evaluate thermocouple error when the reference junction is not fixed at 0°C, one needs limits of error for the Seebeck coefficient (slope of thermocouple voltage vs. temperature curve) for the various thermocouples. Lacking this information, a reasonable approach is to apply the percentage errors, with perhaps 0.25% added on, to the difference in temperature being measured by the thermocouple.

ACCURACY OF THE THERMOCOUPLE VOLTAGE MEASUREMENT

The accuracy of a CR10 voltage measurement is specified as 0.2% (0.1% 0 to 40°C) of the full scale range being used to make the measurement. The actual accuracy may be better than this as it involves a slope error (the error is proportional to the measurement being made though limited by the resolution). The error in the temperature due to inaccuracy in the measurement of the thermocouple voltage is worst at temperature extremes, where a relatively large scale is necessary to read the thermocouple output. For example, assume type K (chromel-alumel) thermocouples are used to measure temperatures at 600°C. The TC output is on the order of 24.9 mV, requiring the +25 mV input range. The accuracy specification of 0.1% FSR is 25 uV which is a temperature error of about 0.60°C. In the environmental temperature range with voltage measured on an appropriate scale, error in

temperature due to the voltage measurements is a few hundredths of a degree.

THERMOCOUPLE POLYNOMIALS - Voltage to Temperature Conversion

NBS Monograph 125 gives high order polynomials for computing the output voltage of a given thermocouple type over a broad range of temperatures. In order to speed processing and accommodate the CR10's math and storage capabilities, 4 separate 6th order polynomials are used to convert from volts to temperature over the range covered by each thermocouple type. Table 13.4-2 gives error limits for the thermocouple linearization functions.

TABLE 13.4-2. Limits of Error on CR10 Thermocouple Output Linearization (Relative to NBS Standards)

TC Type	Range °C		Limits of Error °C
T	**-270 to**	**400**	
	-270 to	-200	±18 @ -270
	-200 to	-100	± 0.08
	-100 to	100	± 0.001
	100 to	400	± 0.015
J	**-150 to**	**760**	± 0.008
	-100 to	300	± 0.002
E	**-240 to**	**1000**	
	-240 to	-130	± 0.4
	-130 to	200	± 0.005
	200 to	1000	± 0.02
K	**-50 to**	**1372**	
	-50 to	950	± 0.01
	950 to	1372	± 0.04

REFERENCE JUNCTION COMPENSATION - Temperature to Voltage

The polynomials used for reference junction compensation (converting reference temperature to equivalent TC output voltage) do not cover the entire thermocouple range. Substantial errors will result if the reference junction temperature is outside of the calibrated range. The ranges covered by these calibrations include the CR10 environmental operating range, so there is no problem when the CR10 is used as the reference junction. External reference junction boxes, however, must also be within these temperature ranges. Temperature difference measurements made outside of the reference temperature range should be made by obtaining the actual temperatures referenced to a junction within the reference temperature range and subtracting. Table 13.4-3 gives the reference temperature ranges covered and the limits of error in the linearizations within these ranges.

Two sources of error arise when the reference temperature is out of range. The most significant error is in the calculated compensation voltage; however, error is also created in the temperature difference calculated from the thermocouple output. For example, suppose the reference temperature for a measurement on a type T thermocouple is 300°C. The compensation voltage calculated by the CR10 corresponds to a temperature of 272.6°C, a -27.4°C error. The type T thermocouple with the measuring junction at 290°C and reference at 300°C would output -578.7 µV; using the reference temperature of 272.6°C, the CR10 calculates a temperature difference of -10.2°C, a -0.2°C error. The temperature calculated by the CR10 would be 262.4°C, 27.6°C low.

TABLE 13.4-3. Reference Temperature Compensation Range and Linearization Error Relative to NBS Standards

TC Type	Range °C	Limits of Error °C
T	-100 to 100	± 0.001
J	-150 to 296	± 0.005
E	-150 to 206	± 0.005
K	-50 to 100	± 0.01

ERROR SUMMARY

The magnitude of the errors described in the previous sections illustrate that the greatest sources of error in a thermocouple temperature measurement are likely to be due to the limits of error on the thermocouple wire and in the reference temperature determined with the built-in thermistor. Errors in the thermocouple and reference temperature polynomials are extremely small, and error in the voltage measurement is negligible.

To illustrate the relative magnitude of these errors in the environmental range, we will take a worst case situation where all errors are maximum and additive. A temperature of 45°C is measured with a type T (copper-constantan) thermocouple, using the ±2.5 mV range. The nominal accuracy on this range is 2.5 µV (0.1% of 2.5 mV), which at 45°C changes the temperature by 0.06ºC. The RTD is 25°C but is

indicating 25.3°C, and the terminal that the thermocouple is connected to is 0.3°C cooler than the RTD.

TABLE 13.4-4. Example of Errors in Thermocouple Temperature

Source	Error °C	% of Total Error	
		1°C Error	1% Slope Error
Reference Temp.	0.6	36.1	69.6
TC Output			
ANSI	1.0	60.1	
0.01 x 20ºC	0.2		23.2
Voltage Measurement	0.06	3.6	7.0
Reference Linearization	0.001	0.1	0.1
Output Linearization	0.001	0.1	0.1
Total Error			
With ANSI error	1.662	100	
Assuming 1% slope error	0.862		100

13.4.2 USE OF EXTERNAL REFERENCE JUNCTION OR JUNCTION BOX

An external junction box is often used to facilitate connections and to reduce the expense of thermocouple wire when the temperature measurements are to be made at a distance from the CR10. In most situations, it is preferable to make the box the reference junction, in which case its temperature is measured and used as the reference for the thermocouples; copper wires are run from the box to the CR10 (Section 3.4). Alternatively, the junction box can be used to couple extension grade thermocouple wire to the thermocouples being used for measurement, and the CR10 panel used as the reference junction. Extension grade thermocouple wire has a smaller temperature range than standard thermocouple wire, but meets the same limits of error within that range. The only situation where it would be necessary to use extension grade wire instead of an external measuring junction is where the junction box temperature is outside the range of reference junction compensation provided by the CR10. This is only a factor when using type K thermocouples, where the upper limit of the reference compensation linearization is 100°C and the upper limit of the extension grade wire is 200°C. With the other types of thermocouples, the reference compensation range equals or is greater than the extension wire range. In any case, errors can arise if temperature gradients exist within the junction box.

Figure 13.4-4 illustrates a typical junction box. Terminal strips will be a different metal than the thermocouple wire. Thus, if a temperature gradient exists between A and A' or B and B', the junction box will act as another thermocouple in series, creating an error in the voltage measured by the CR10. This thermoelectric offset voltage is a factor whether or not the junction box is used for the reference. It can be minimized by making the thermal conduction between the two points large and the distance small. The best solution in the case where extension grade wire is being connected to thermocouple wire would be to use connectors which clamped the two wires in contact with each other.

An external reference junction box must be constructed so that the entire terminal area is very close to the same temperature. This is necessary so that a valid reference temperature can be measured, and to avoid a thermoelectric offset voltage which will be induced if the terminals at which the thermocouple leads are connected (points A and B in Figure 13.4-4) are at different temperatures. The box should contain elements of high thermal conductivity, which will act to rapidly remove any thermal gradients to which the box is subjected. It is not necessary to design a constant temperature box; it is desirable that the box respond slowly to external temperature fluctuations.

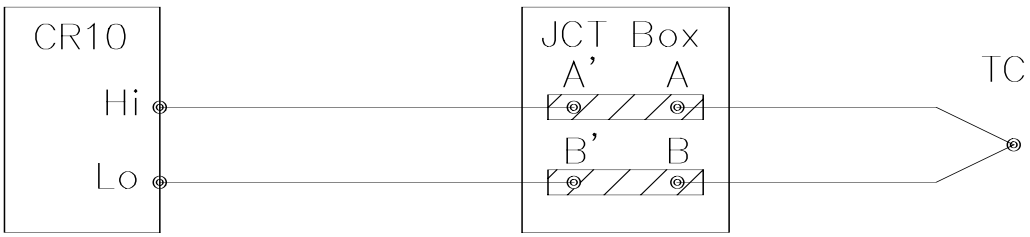


FIGURE 13.4-2. Diagram of Junction Box

Radiation shielding must be provided when a junction box is installed in the field. Care must also be taken that a thermal gradient is not induced by conduction through the incoming wires. The CR10 can be used to measure the temperature gradients within the junction box.

13.5 BRIDGE RESISTANCE MEASUREMENTS

There are 6 bridge measurement instructions included in the standard CR10 software. Figure 13.5-1 shows the circuits that would typically be measured with these instructions. In the diagrams, the resistors labeled R_s would normally be the sensors and those labeled R_f would normally be fixed resistors. Circuits other than those diagrammed could be measured, provided the excitation and type of measurements were appropriate.

With the exception of Instructions 4 and 8, which apply an excitation voltage then wait a specified time before making a measurement, all of the bridge measurements make one set of measurements with the excitation as programmed and another set of measurements with the excitation polarity reversed. The error in the two measurements due to thermal emfs can then be accounted for in the processing of the measurement instruction. The excitation is switched on 450 µs before the integration portion of the measurement starts and is grounded as soon as the integration is completed. Figure 13.5-2 shows the excitation and measurement sequence for Instruction 6, a 4 wire full bridge. Excitation is applied separately for each phase of a bridge measurement. For example, in Instruction 6, as shown in Figure 13.5-2, excitation is switched on for the 4 integration periods and switched off between integrations.

Instruction 8 measurement sequence consists of applying a single excitation voltage, delaying a specified time, and making a differential voltage measurement. If a delay of 0 is specified, the inputs for the differential measurement are not switched for a second integration as is normally the case (Section 13.2). The result stored is the voltage measured. Instruction 8 does not have as good resolution or common mode rejection as the ratiometric bridge measurement instructions. It does provide a very rapid means of making bridge measurements as well as supplying excitation to circuitry requiring differential measurements. This instruction does not reverse excitation. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

The output of Instruction 8 is simply the voltage measurement. When 8 is used to measure a full bridge (same connections as Instruction 6 in Figure 13.5-1), the result is V_1 which equals V_x $(R_3/(R_3+R_4) - R_2/(R_1+R_2))$. (In other words, to make the output the same as Instruction 6, use a factor of $1000/V_x$ in the multiplier.)

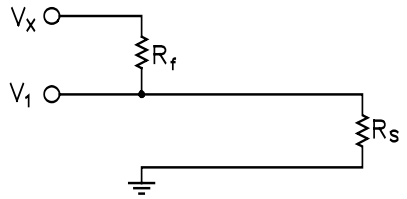
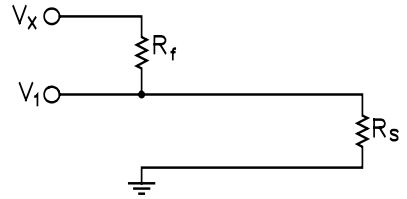
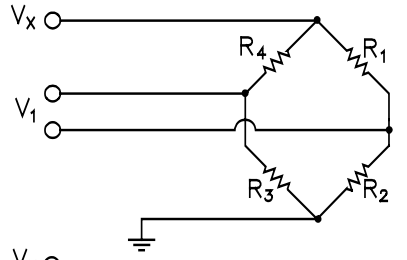
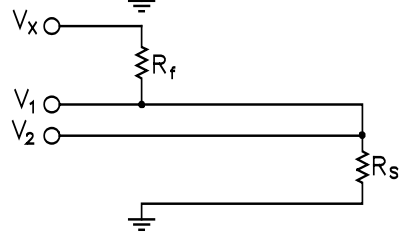
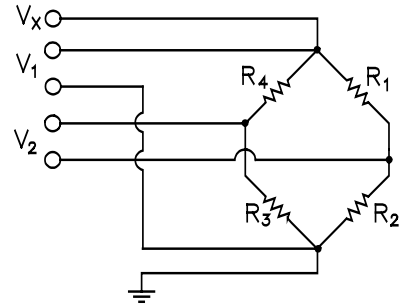
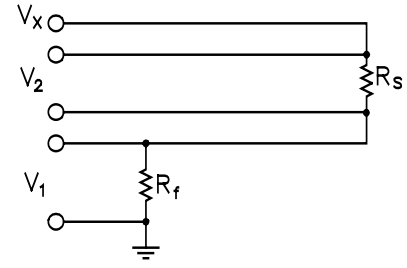
INSTR. #	DIAGRAM	DESCRIPTION	RESULT = X
4		DC HALF BRIDGE WITH USER ENTERED SETTLING TIME	$X = V_1 = V_x \frac{R_s}{R_s + R_f}$
5		AC HALF BRIDGE EXCITATION ALTERNATES POLARITY FOR ION DEPOLARIZATION	$X = \frac{V_1}{V_x} = \frac{R_s}{R_s + R_f}$
6		4 WIRE FULL BRIDGE	$X = 1000 \frac{V_1}{V_x} = 1000 \left(\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2}\right)$
7		3 WIRE HALF BRIDGE	$X = \frac{2V_2 - V_1}{V_x - V_1} = \frac{R_s}{R_f}$
9		6 WIRE FULL BRIDGE WITH EXCITATION LEAD COMPENSATION	$X = 1000 \frac{V_2}{V_1} = 1000 \left(\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2}\right)$ (V_1 ON 2.5 V RANGE)
9		4 WIRE HALF BRIDGE	$X = \frac{V_2}{V_1} = \frac{R_s}{R_f}$ (V_1 NOT ON 2.5 V RANGE)

FIGURE 13.5-1. Circuits Used with Instructions 4-9

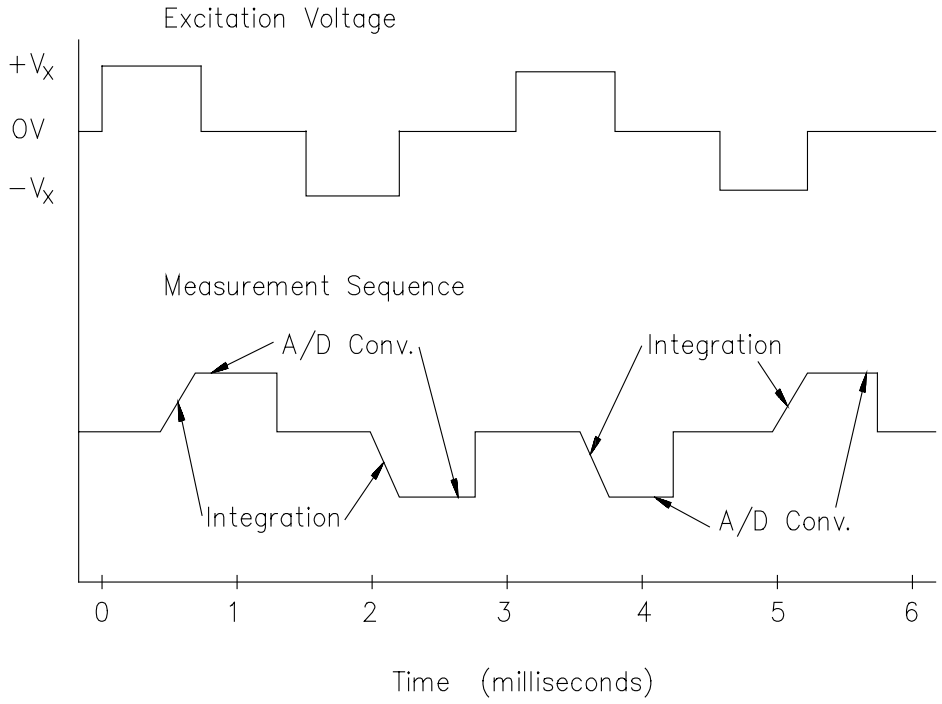


FIGURE 13.5-2. Excitation and Measurement Sequence for 4 Wire Full Bridge

TABLE 13.5-1. Comparison of Bridge Measurement Instructions

Instr. #	Circuit	Description
4	DC Half Bridge	The delay parameter allows the user entered settling time compensate for capacitance in long lead lengths. No polarity reversal. One single-ended measurement. Measured voltage output.
5	AC Half Bridge	Rapid reversal of excitation polarity for ion depolarization. One single-ended measurement at each excitation polarity. Ratiometric output.
6	4 Wire Full Bridge	One differential measurement at each excitation polarity. Ratiometric output.
7	3 Wire Half Bridge	Compensates for lead wire resistance, assuming resistance is same in both wires. Two single-ended measurements at each excitation polarity. Ratiometric output.
8	Differential Measurement with Excitation	Makes a differential measurement without reversing excitation polarity. Used for fast measurements on load cells, PRTs etc. Resolution and common mode rejection worse than 6 if used with delay = 0. Measured voltage output.
9	6 Wire Full Bridge or 4 Wire Half Bridge	Compensates for lead wire resistance. Two differential measurements at each excitation polarity. Ratiometric output.

Calculating the actual resistance of a sensor which is one of the legs of a resistive bridge usually requires the use of one or two Processing Instructions in addition to the bridge measurement instruction. Instruction 59 takes a value, X, in a specified input location and computes the value MX/(1-X), where M is the multiplier and stores the result in the original location. Instruction 42 computes the reciprocal of a value in an input location. Table 13.5-2 lists the instructions used to compute the resistance of any single resistor shown in the diagrams in Figure 13.5-1, provided the values of the other resistors in the bridge circuit are known.

TABLE 13.5-2. Calculating Resistance Values from Bridge Measurement

Instr.	Result	Instr.#	Multiplier;	Offset
4	$X = V_x(R_s/(R_s+R_f))$			
	$R_s = R_f \frac{X/V_x}{1-X/V_x}$	**4** **59**	$1/V_x$ R_f	0
	$R_f = \frac{1}{((X/V_x)/(1-X/V_x))/R_s}$	**4** **59** **42**	$1/V_x$ $1/R_s$	0
5	$X = R_s/(R_s+R_f)$			
	$R_s = R_f \frac{X}{1-X}$	**5** **59**	1 R_f	0
	$R_f = \frac{1}{(X/(1-X))/R_s}$	**5** **59** **42**	1 $1/R_s$	0 0
6,8,9*	$X = 1000\,[R_3/(R_3+R_4)-R_2/(R_1+R_2)]$	*used for full bridge		
	$R_1 = \frac{1}{(X_1/(1-X_1))/R_2}$ where $X_1 = -X/1000 + R_3/(R_3+R_4)$	**6** or **9** **8** **6**,**8**, or **9** **59** **42**	-0.001; $1/V_x$ $1/R_2$	 $R_3/(R_3+R_4)$
	$R_2 = R_1(X_2/(1-X_2))$ where $X_2 = X_1$	**6** or **9** **59**	-0.001 R_1	$R_3/(R_3+R_4)$
	$R_3 = R_4(X_3/(1-X_3))$ where $X_3 = X/1000 + R_2/(R_1+R_2)$	**6** or **9** **59**	0.001 R_4	$R_2/(R_1+R_2)$
	$R_4 = \frac{1}{(X_4/(1-X_4))/R_3}$ where $X_4 = X_3$	**6** or **9** **59** **42**	0.001 $1/R_3$	$R_2/(R_1+R_2)$
7&9*	$X = R_s/R_f$	* used as half bridge		
	$R_s = R_fX$	**7** or **9**	R_f	0

$R_f = R_s/X$	**7** or **9** **42**	$1/R_s$	0

13.6 RESISTANCE MEASUREMENTS REQUIRING AC EXCITATION

Some resistive sensors require AC excitation. These include the 207 Relative Humidity Probe, soil moisture blocks, water conductivity sensors, and wetness sensing grids. The use of DC excitation with these sensors can result in polarization, which will cause an erroneous measurement, and may shift the calibration of the sensor and/or lead to its rapid decay.

The AC half bridge Instruction 5 (incorporated into the 207 relative humidity measurement Instruction 12) reverses excitation polarity to provide ion depolarization and, in order to minimize the time excitation is on, grounds the excitation as soon as the signal is integrated (Figure 13.6-1). The slow integration time should never be used with a sensor requiring AC excitation because it results in the excitation lasting about 1.5 times as long, allowing polarization to affect the measurement.

INFLUENCE OF GROUND LOOP ON MEASUREMENTS

When measuring soil moisture blocks or water conductivity, the potential exists for a ground loop which can adversely affect the measurement. This ground loop arises because the soil and water provide an alternate path for the excitation to return to CR10 ground, and can be represented by the model diagrammed in Figure 13.6-2.

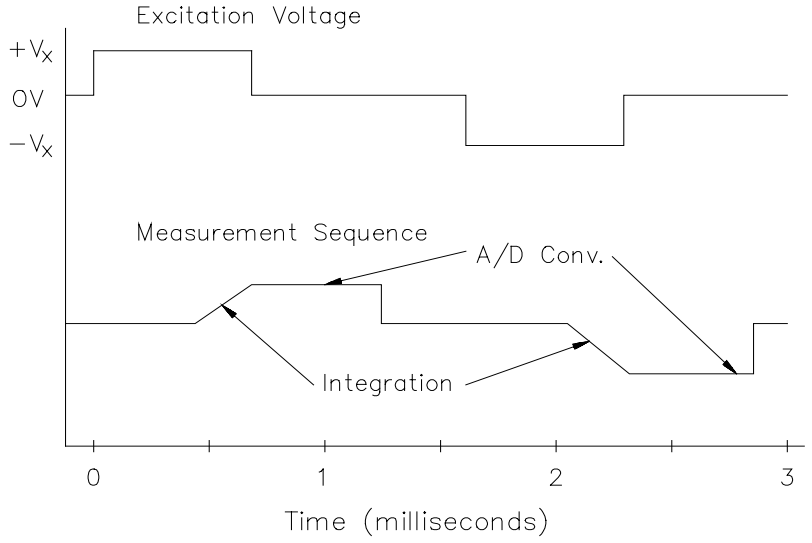


FIGURE 13.6-1. AC Excitation and Measurement Sequence for AC Half Bridge

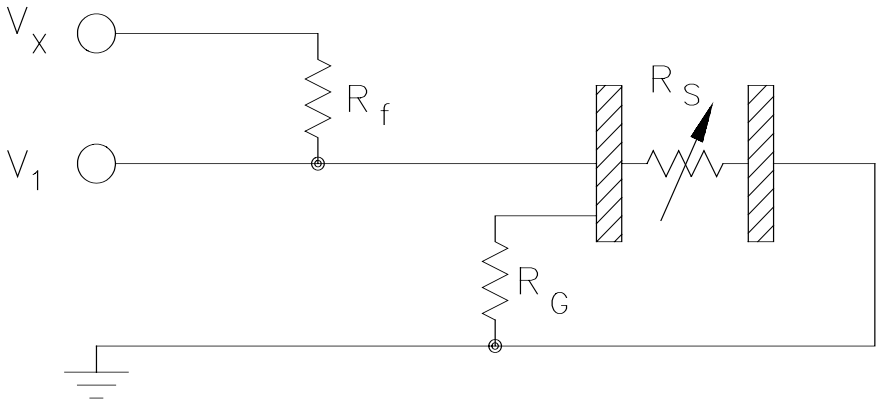


FIGURE 13.6-2. Model of Resistive Sensor with Ground Loop

In Figure 13.6-2, V_x is the excitation voltage, R_f is a fixed resistor, R_s is the sensor resistance, and R_G is the resistance between the excited electrode and CR10 earth ground. With R_G in the network, the measured signal is:

$$V_1 = V_x \frac{R_s}{(R_s+R_f) + R_sR_f/R_G} \quad [13.6\text{-}1]$$

R_sR_f/R_G is the source of error due to the ground loop. When R_G is large the equation reduces to the ideal. The geometry of the electrodes has a great effect on the magnitude of this error. The Delmhorst gypsum block used in the 227 probe has two concentric cylindrical electrodes. The center electrode is used for excitation; because it is encircled by the ground electrode, the path for a ground loop through the soil is greatly reduced. Moisture blocks which consist of two parallel plate electrodes are particularly susceptible to ground loop problems. Similar considerations apply to the geometry of the electrodes in water conductivity sensors.

The ground electrode of the conductivity or soil moisture probe and the CR10 earth ground form a galvanic cell, with the water/soil solution acting as the electrolyte. If current was allowed to flow, the resulting oxidation or reduction would soon damage the electrode, just as if DC excitation was used to make the measurement. Campbell Scientific probes are built with series capacitors in the leads to block this DC current. In addition to preventing sensor deterioration, the capacitors block any DC component from affecting the measurement.

13.7 CALIBRATION PROCESS

The CR10 makes voltage measurements by integrating the input signal for a fixed time and then holding the integrated value for the analog to digital (A/D) conversion. The A/D conversion is made by a 13 bit approximation using a digital to analog converter (DAC). The result from the approximation is DAC counts, which are multiplied by coefficients to obtain millivolts (mV). There are 10 calibration coefficients, one for each of the 5 gain ranges for the fast and slow integration times.

The CR10 has an internal calibration function that feeds positive and negative voltages through the amplifiers and integrator and calculates new calibration coefficients. By adjusting the calibration coefficients the accuracy of the voltage measurements is maintained over the -25 to +50°C operating range of the CR10. Calibration is executed under four conditions:

1. When the CR10 is powered up.
2. Automatically when Instruction 24 is not contained in a program table.
3. When the watchdog resets the processor.
4. When the calibration instruction, Instruction 24, is executed.

AUTOMATIC CALIBRATION SEQUENCE

The primary advantage of automatic calibration is that the CR10 is constantly calibrated without user programming. The CR10 defaults to automatic calibration when Instruction 24 is not contained in a program table.

Every 8 seconds one part of a 22 part calibration sequence is performed. Program execution is interrupted (5.4 - 21.4 ms), when necessary, for each part of the calibration. Every 2.9 minutes (8 seconds * 22) ten calibration coefficients are calculated. The calculated coefficients are multiplied by 1/5, and then added to 4/5 times the existing coefficients. Averaging is done as a safeguard against coefficients calculated from a noisy measurement.

The above weighting of the newly calculated coefficients results in a 15 minute time constant (see Instruction 58) in the response of the calibration to step changes affecting the calibration coefficients (primarily temperature). For most environmental applications a 15 minute time constant is acceptable. The automatic calibration may result in the calibration coefficients not being optimum for applications that subject the CR10 to extreme temperature gradients.

Automatic calibration extends the processing time 5.4 to 21.4 ms when it is executed (every 8

seconds). If the processing time exceeds the execution interval the CR10 finishes processing the table and awaits the next occurrence of the execution interval before initiating the table. At the fastest execution interval of 1/64 (0.0156) second the program table WILL be overrun by the automatic calibration. If an overrun occurs every time calibration is executed, then 1 execution is skipped for every 512 times that the program table is executed. If the measurements are being averaged, the effect of the overrun is negligible. Program table overruns are indicated by the appearance of two decimals on either side of the sixth digit on the CR10KD and are also stored in memory (Section 1.7).

INSTRUCTION 24 CALIBRATION

The alternative to automatic calibration is the use of Instruction 24, the calibration instruction. Instruction 24 implements a complete calibration which occurs ONLY when EXECUTED by a program table. Instruction 24 calibration is the average of 10 calibrations, and takes approximately 2.8 seconds to complete. Automatic calibration is disabled when a program is compiled that contains Instruction 24.

Instruction 24 calibration, as opposed to automatic calibrations, may be advantageous in applications where: 1) the CR10 is exposed to extreme thermal gradients, or 2) automatic calibration would interfere with the desired sampling rate, and the ambient temperature is stable enough to allow calibration at specific points during program execution.

Calibration coefficients are replaced each time that Instruction 24 is executed. Unlike automatic calibration, there is no time constant for the coefficients to respond in changes to calibration. Instruction 24 calibration ensures that the coefficients are optimum at the time that the instruction is executed. For example, consider a CR10 mounted under the dash of an automobile, where temperature could easily change 50 degrees. Temperature changes affect the measurement circuitry which must be compensated for by calculating new coefficients. Each time Instruction 24 is executed a new set of calibration coefficients is calculated based on the measurements made at that time.

Calibration at a certain point during program execution may be advantageous for some applications. For example, suppose Table 2 has an execution time of 15.6 ms, but only executes when flag 1 is set. Table 1 has a 5 minute execution time which makes a temperature measurement, and sets flag 1 if the temperature exceeds a fixed value. To prevent overrun errors which would occur in Table 2 if the automatic calibration was used, Instruction 24 could be executed before the temperature measurement was made by Table 1.

Instruction 24 also has an option to store the results of the automatic calibration in Input Storage. This can be used to detect hardware problems. If -99999 appears in any of the 19 input locations, the CR10 has a hardware problem or needs factory calibration.

SECTION 14. INSTALLATION AND MAINTENANCE

14.1 PROTECTION FROM THE ENVIRONMENT

The normal environmental variables of concern are temperature and moisture. The standard CR10 is designed to operate reliably from -25 to +50°C (-55° to +85°C, optional). Internal moisture is eliminated by sealing the module at the factory with three packets of silica gel (0.75 g each) inside. The desiccant is replaced whenever the CR10 is repaired at Campbell Scientific. The module should not be opened by the user except for the infrequent purpose of PROM replacement (refer to Appendix G). Repeated disassembly/assembly of the CR10 will degrade the seal, leading to potential moisture problems. Extra desiccant should also be placed in the enclosure to prevent corrosion on the Wiring Panel terminals and CR10/Wiring Panel connections.

Campbell Scientific offers two enclosures for housing a CR10 and peripherals. The fiberglass enclosures are classified as NEMA 4X (water-tight, dust-tight, corrosion-resistant, indoor and outdoor use). A 1.25" diameter entry/exit port is located at the bottom of the enclosure for routing cables and wires. The enclosure door can be fastened with the hasp for easy access, or with the two supplied screws for more permanent applications. The white plastic inserts at the corners of the enclosure must be removed to insert the screws. Both enclosures are white for reflecting solar radiation, thus reducing the internal enclosure temperature.

The Model ENC 12/14 fiberglass enclosure houses the CR10, power supply, and one or more peripherals. Inside dimensions of the ENC 12/14 are 14" x 12" x 5.5", outside dimensions are 18" x 13.5" x 8.13" (with brackets); weight is 11.16 lbs.

The Model ENC 16/18 fiberglass enclosure houses the CR10 Power Supply, and several peripherals. Inside dimensions of the ENC 16/18 are 16" x 18" x 9", outside dimensions are 21.75" x 20.0" x 11.0" (with brackets); weight is 17.2 lbs.

14.2 POWER REQUIREMENTS

The CR10 operates at a nominal 12 VDC. Below 9.6 or above 16 volts the CR10 does not operate properly.

The CR10 is diode protected against accidental reversal of the positive and ground leads from the battery. Input voltages in excess of 18 V may damage the CR10 and/or power supply. A transzorb provides transient protection by limiting voltage at approximately 20 V.

CAUTION: The metal surfaces of the Wiring Panel and mounting bracket are at the same electrical ground as the power supply. Caution must be exercised when connecting power directly to the Wiring Panel's 12 V and ground terminals. Connect the plus (+) side of the supply first, keeping the minus (-) side away from the Wiring Panel. Once the plus side is secured, connect the power return.

TABLE 14.2-1. Typical Current Drain for Common CR10 Peripherals

	Typical Current Drain (mA)	
Peripheral	**Quiescent**	**Active**
RF100 VHF 5 Watt Radio	30	1730
RF200 UHF 4 Watt Radio	30	1530
RF95 RF Modem	1.4	30
SM192/SM716 Storage Module	0.25	3
CSM1	0.5	18
CR10KD Keyboard/Display	<1	0.3 (*6 Mode)
DC112 Phone Modem	<1	45
SDM-AO4	10.5	--
SDM-CD16	6.0	11/LED lit
SDM-INT8	0.4	6.5
SDM-SW8A	3	6

System operating time for the batteries can be determined by dividing the battery capacity (amp-hours) by the average system current drain. The CR10 draws <1 mA in the quiescent state, 13 mA while processing, and 46 mA during an analog measurement; the length of operating time for each datalogger instruction is listed in the programming section. Typical current requirements for common CR10 peripherals are given in Table 14.2-1.

14.3 CAMPBELL SCIENTIFIC POWER SUPPLIES

The PS12 Power Supply is available from Campbell Scientific with either alkaline or lead acid batteries, the PS12ALK and PS12LA, respectively. The PS512M is also a lead acid supply with two 9-pin null modem ports for communication modems, see Section 14.3.3. The PS12ALK has 8 D cell alkaline batteries, the PS12LA has a rechargeable lead acid battery. The alkaline batteries are discarded after use. The lead acid batteries should be float charged with either AC power or a solar panel. The lead acid battery supplies power during a power failure or in times of low charge with a solar panel.

The CH12R and CH512R contain the same circuitry as the PS12LA and PS512M, respectively. They are used to float charge an external 12 VDC Yuasa battery using AC or solar power. No internal batteries are contained in the CH12R and CH512R. Their operation, however, is identical to that of the PS12LA and PS512M. Other power supply options are connecting a 12 volt battery directly to the CR10, Section 14.5, or supplying power from a vehicle, Section 14.6.

The PS12 Power Supply provides 12 volts, regulates incoming AC or DC power, limits current from the battery, and provides circuitry to connect an external 12 volt battery. The terminals on the PS12 are exposed by unscrewing the two set screws, as shown in Figure 4.3-1.

The two 12 volt and two ground terminals are for supplying power to the datalogger, or other 12 volt devices.

The two terminals, labeled CHG, are for connecting a 20 VDC adapter or solar panel to charge lead acid batteries.

The charge input can be either AC or DC, and it does not matter which terminal is positive or negative. The voltage input must be within 16 to 26 VDC, or 16 to 26 VAC RMS.

The ON-OFF switch controls power to the 12 V ports. Charging of lead acid batteries still occurs when the switch is off. The red charge light is on when a charging source is connected to the power supply.

The connectors labeled INT and EXT are for connecting the internal (power supply) battery and an external battery, respectively. A five foot cable, with connector, is included with the power supply for connection to an external battery. This is commonly used for supplying power to the datalogger while changing power supply batteries.

A thermal fuse in the power circuit limits source current. If excessive current is drawn, the fuse gets hot, increases in resistance, and limits current. When the problem is fixed, the fuse cools and the resistance decreases, eventually allowing current to pass. When excessive current is drawn due to shorting the power leads to the Wiring Panel, allow 10 to 15 seconds for the fuse to cool before connecting power.

14.3.1 PS12ALK ALKALINE POWER SUPPLY

The PS12ALK utilizes 8 alkaline D cells mounted in place of the lead acid battery shown in Figure 14.3-1. The PS12ALK can also be used with a lead acid battery connected to the external battery port, in this case the alkaline batteries act as a backup.

Before installing the alkaline batteries, connect all necessary sensor leads, control lines, and power leads. The CR10 can be turned on and off with the switch on the PS12ALK.

To replace the batteries without losing the datalogger program and data: 1) do not turn the power switch off, 2) connect an external battery to the port labeled EXT with the supplied 6 foot cable, 3) remove the old batteries, 4) replace with new alkaline D cell batteries, and 5) remove the external battery.

A fresh set of eight alkaline D cells has 12.4 volts and a nominal rating of 7.5 amp-hours at 20°C. The amp-hour rating decreases with temperature as shown in Table 14.3-1. Datalogger Instruction 10 can be used to

monitor battery voltage. Replace the alkaline cells before the CR10 battery voltage drops below 9.6 V.

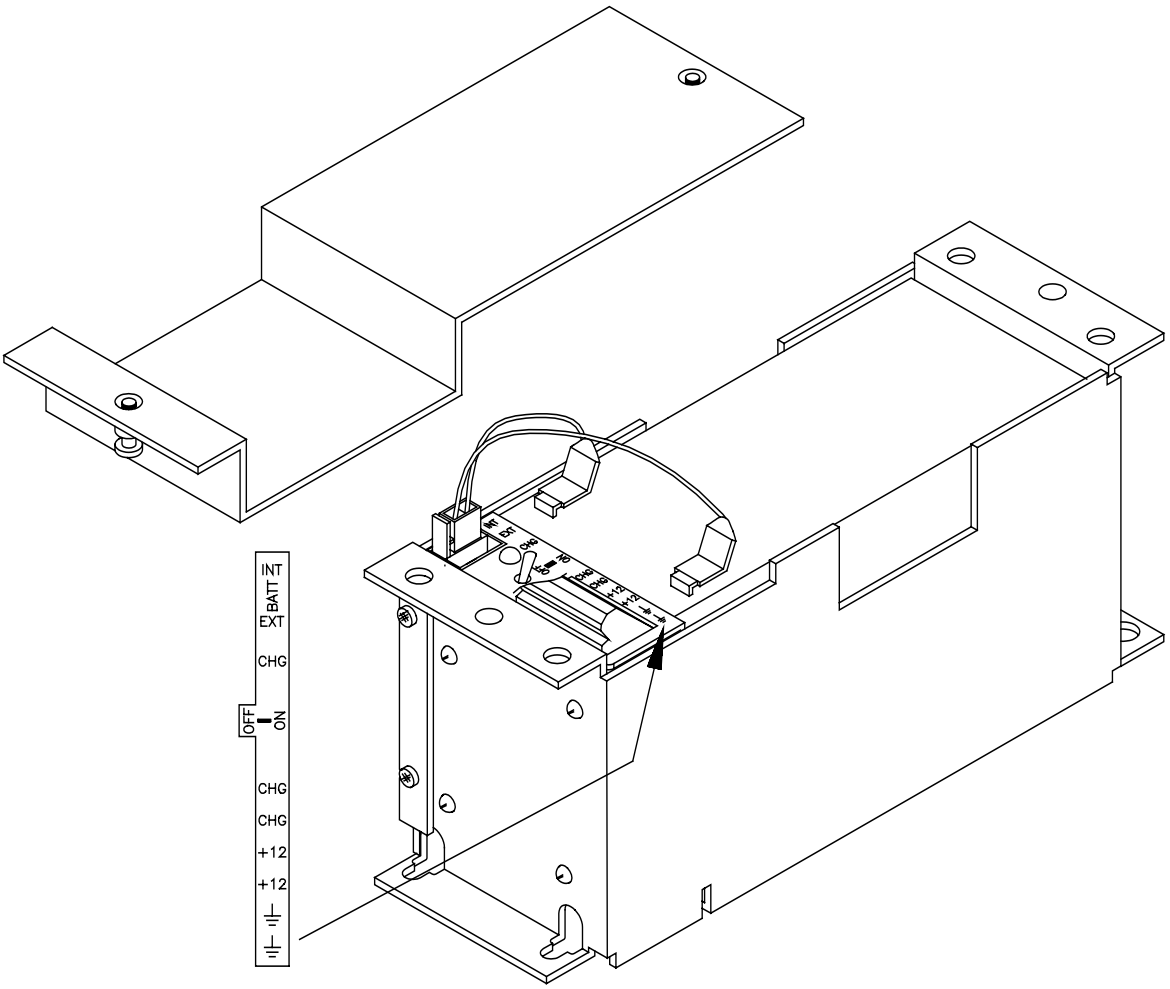


FIGURE 14.3-1. PS12 12 Volt Power Supply and Charging Regulator

TABLE 14.3-1. Typical Alkaline Battery Service and Temperature

Temperature (°C)	% of 20°C Service
20 - 50	100
15	98
10	94
5	90
0	86
-10	70
-20	50
-30	30

NOTE: This data is based on one "D" cell under conditions of 50 mA current drain with a 30 ohm load. As the current drain decreases, the percent service improves for a given temperature.

The five foot external battery cable can be connected to a large capacity (amp-hour) battery to power the system for an extended period of time. The red and black leads connect to the positive and negative battery posts, respectively. The alkaline cells act as a power backup in this case. A diode in the PS12ALK isolates the lead acid supply from the alkaline batteries.

14.3.2 PS12LA LEAD ACID POWER SUPPLY

The PS12LA power supply includes a 12 V, 7.0 amp-hour lead acid battery, a AC transformer (18 V), and a temperature compensated charging circuit with a charge indicating diode. An AC transformer or solar panel should be connected to the PS12 at all times. The charging source powers the CR10 while float charging the lead acid batteries. The internal lead acid battery powers the datalogger if the

charging source is interrupted. The PS12LA specifications are given in Table 14.3-2.

The two leads from the charging source can be inserted into either of the CHG ports, polarity doesn't matter. A transzorb provides transient protection to the charging circuit. A sustained input voltage in excess of 40 V will cause the transzorb to limit voltage.

Some solar panels are supplied with a connector, this connector must clipped off so the two wires can be inserted into the two terminal ports. It is recommended that these two leads be stripped and tinned.

The red light (LED) on the PS12LA is on when a charging source is connected to the PS12LA CHG ports. The switch turns power on and off to the 12 V ports, battery charging still occurs when the switch is off.

> **CAUTION:** Switch the power to "off" before disconnecting or connecting the power leads to the Wiring Panel. The Wiring Panel and PS12LA are at power ground. If 12 V is shorted to either of these, excessive current will be drawn until the thermal fuse opens.

The external port, labeled EXT, is not meant to be used with the PS12LA. The primary power source is the charging source, and the secondary power source is the internal lead acid battery. Connecting a lead acid battery to the external source is the same as connecting two lead acid batteries in parallel, causing one battery to drop voltage and the other to raise voltage. Alkaline batteries connected to the external port must have a diode in series to block charging which would cause an explosion. (The PS12ALK battery pack has this diode.)

Monitor the power supply using datalogger Instruction 10. Users are strongly advised to incorporate this instruction into their data acquisition programs to keep track of the state of the power supply. If the system voltage level consistently decreases through time, some element(s) of the charging system has failed. Instruction 10 measures the voltage at the 12 V port, not the voltage of the lead acid battery. External power sources must be disconnected from the CR10 and charging circuit in order to measure the actual lead acid battery voltage.

TABLE 14.3-2. PS12LA Battery and AC Transformer Specifications

Lead Acid Battery	
Battery Type	Yuasa NA 7-12
Float Life @ 25°C	5 years typical
Capacity	7.0 amp-hour
Shelf Life, full charge	Check twice yearly
Charge Time (AC Source)	40 hr full charge, 20 hr 95% charge
AC Transformer	
Input:	120 VAC, 60 Hz
Isolated Output:	18 VDC @ 1.11 A max.

There are inherent hazards associated with the use of sealed lead acid batteries. Under normal operation, lead acid batteries generate a small amount of hydrogen gas. This gaseous by-product is generally insignificant because the hydrogen dissipates naturally before build-up to an explosive level (4%) occurs. However, if the batteries are shorted or overcharging takes place, hydrogen gas may be generated at a rate sufficient to create a hazard. Campbell Scientific makes the following recommendations:

1. A CR10 equipped with standard lead acid batteries should NEVER be used in applications requiring intrinsically safe equipment.
2. A lead acid battery should not be housed in a gas-tight enclosure.

14.3.3 PS512M VOLTAGE REGULATOR WITH NULL MODEM PORTS

The PS512M 12 Volt Lead Acid Power Supply with Charging Regulator and Null Modem Ports is used when 5 volts is needed to power external modems besides the capabilities of the PS12LA. The PS512M supplies 5 volts to pin 1 of the 9 pin null modem ports, otherwise the capabilities and functions are identical to the PS12LA. A common use for the PS512M is in radiotelemetry networks. The PS12LA cannot be modified to a PS512M.

The maximum current drain on the 5 volt supply of the PS512M is 150 milliamps.

14.4 SOLAR PANELS

Auxiliary photovoltaic power sources may be used to maintain charge on lead acid batteries.

When selecting a solar panel, a rule-of-thumb is that on a stormy overcast day the panel should provide enough charge to meet the system current drain (assume 10% of average annual global radiation, kW/m^2). Specific site information, if available, could strongly influence the solar panel selection. For example, local effects such as mountain shadows, fog from valley inversion, snow, ice, leaves, birds, etc. shading the panel should be considered.

Guidelines are available from the Solarex Corporation for solar panel selection called "DESIGN AIDS FOR SMALL PV POWER SYSTEMS". It provides a method for calculating solar panel size based on general site location and system power requirements. If you need help in determining your system power requirements contact Campbell Scientific's Marketing Department.

14.5 DIRECT BATTERY CONNECTION TO THE CR10 WIRING PANEL

For some applications, size restrictions or other operational considerations may preclude the use of Campbell Scientific power supply options. In these cases the power supply may be connected directly to the wiring panel. Any 9.6 to 18 VDC supply may be connected to the 12 V and G terminals on the wiring panel. The metal surfaces of the wiring panel and mounting bracket are at power ground. Make connections to the wiring panel first and then to the power supply. If the power supply must be connected first, connect the positive to the wiring panel before the ground to avoid shorting to the wiring panel or mounting bracket.

14.6 VEHICLE POWER SUPPLY CONNECTIONS

If a CR10 is to be powered from the 12 Volts of a motor vehicle, a second 12 V supply is also required at the time of vehicle start-up. When the starting motor of a vehicle with a 12 V electrical system is engaged, the voltage drops considerably below the nominal 12 V, which would cause the CR10 to malfunction every time the vehicle is started. The second 12 V supply prevents this malfunction. Figure 14.6-1 shows the general case for connecting the two supplies to the Wiring Panel. The diode allows the vehicle to power the CR10 without the second supply attempting to power the vehicle. To reduce the potential for ground reference errors in measurements, the ground lead should be 16 AWG or larger.

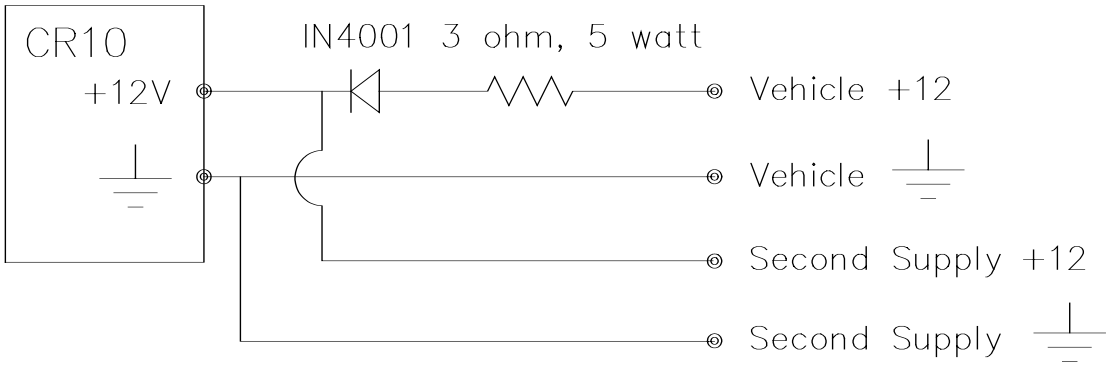


FIGURE 14.6-1. Connecting to Vehicle Power Supply

14.7 GROUNDING

14.7.1 PROTECTION FROM LIGHTNING

Primary lightning strikes are those where lightning hits the datalogger or sensors directly. Secondary strikes occur when the lightning strikes somewhere near the system and induces a voltage in the wires. The purpose of an earth ground is to minimize damage to the system by providing a low resistance path around the system to a point of low potential. Campbell Scientific recommends that all dataloggers in use in the field be earth grounded. All components of the system (datalogger, sensors, external power supplies, mounts, housings, etc.) should be referenced to one common earth ground.

Every terminal on the Wiring Panel, with the exception of ground (G) and analog ground (AG) terminals are spark gapped. The spark gaps will fire at 150 V and the current will be diverted to ground. As shown in Figure 14.7-1, the power ground and analog ground are independent lines until joined inside the CR10. The fuse shown in Figure 14.7-1 (located on the underside of the Wiring Panel) is a 30 AWG wire, equivalent to a conventional 5 Amp fuse. It will blow if a sufficient transient comes in on the G or AG lines, at which time the current is directed away from the CR10 through the diodes. The fuse may be replaced by soldering another 30 AWG wire to the soldering pads provided.

A modem/phone line connected to the Wiring Panel provides another path for transients to enter and damage the CR10. Campbell Scientific's DC112 phone modem has spark gaps on the phone lines. A 12 AWG wire should be run from the modem ground terminal to the earth ground. Additional protection is provided by the ground (Pin 2) of the 9 pin Serial I/O which is tied to power ground on the Wiring Panel.

The transient protection designed into Campbell Scientific's equipment is meaningless if a good system earth ground is not provided. It is the users responsibility to provide this earth ground.

In laboratory applications, locating a stable earth ground is not always obvious. In older buildings, new cover plates on old AC sockets may indicate that a safety ground exists when in fact the socket is not grounded. If a safety ground does exist, it is good practice to verify that it carries no current. If the integrity of the AC power ground cannot be verified, it is better to ground the system to a massive metal object such as a steel water pipe.

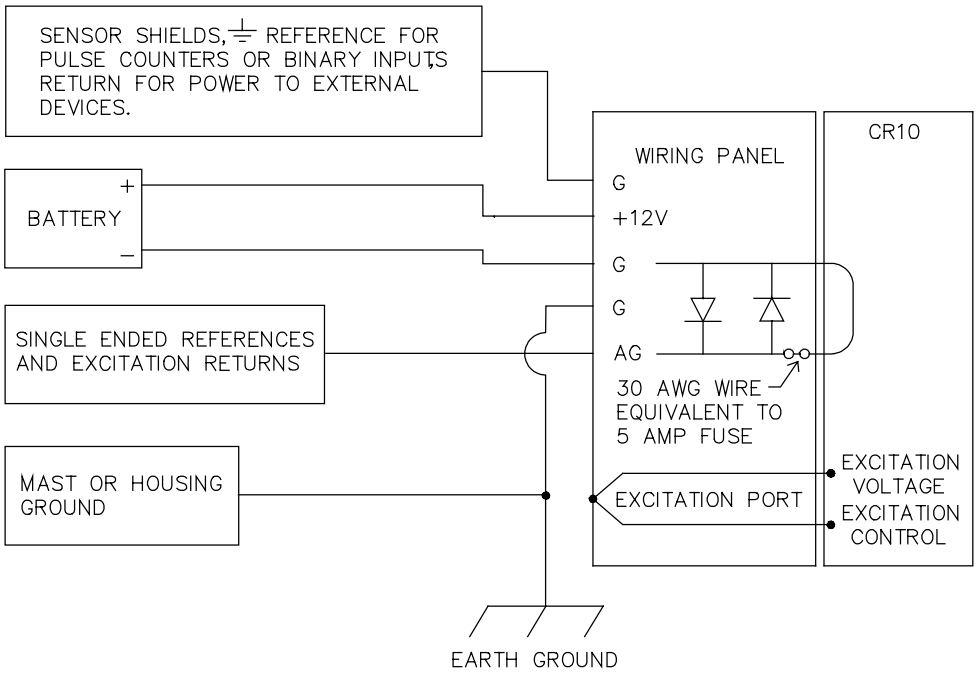


FIGURE 14.7-1. Wiring Panel Grounding Diagram and Excitation Control

In the field, an earth ground may be created through a grounding rod. A 12 AWG or larger wire should be run between a Wiring Panel power ground (G) terminal and the earth ground. Campbell Scientific's CM10 and CM6 Tripods come complete with ground and lightning rods, grounding wires, and appropriate ground wire clamps.

14.7.2 EFFECT OF GROUNDING ON MEASUREMENTS: COMMON MODE RANGE

The common mode range is the voltage range, relative to the CR10 ground, within which both inputs of a differential measurement must lie in order for the differential measurement to be made. Common mode range for the CR10 is ±2.5 V. For example, if the high side of a differential input is at 2 V and the low side is at 0.5 V relative to CR10 ground, a measurement made on the ±2.5 V range would indicate a signal of 1.5 V. However, if the high input changed to 3 V, the common mode range is exceeded and the measurement cannot be made.

Common mode range may be exceeded when the CR10 is measuring the output from a sensor which has its own grounded power supply and the low side of the signal is referenced to power ground. If the CR10 ground and the sensor ground are at sufficiently different potentials, the signal will exceed the common mode range. To solve this problem, the sensor power ground and the CR10 ground should be connected, creating one ground for the system.

In a laboratory application, where more than one AC socket may be used to power various sensors, it is not always safe to assume that the power grounds are at the same potential. To be safe, the ground of all the AC sockets in use should be tied together with a 12 AWG wire.

14.8 WIRING PANEL

The purpose of the Wiring Panel is to provide transient protection, improve excitation voltage accuracy, and make convenient, positive connections of power, sensors, and peripherals to the CR10 (refer to Figure 14.7-1). Wiring Panel transient protection is discussed in Section 14.7.

The Wiring Panel carries two lines between the CR10 and each excitation port. One line is for excitation voltage, the other is for feedback control of the voltage. The feedback line is required to compensate for line losses between the CR10 and the excitation port on the Wiring Panel (see Figure 14.7-1).

Two 5 V output terminals are available on the Wiring Panel for powering 5 V peripherals. The most common use of these terminals is to switch the 5 V to a relay coil through a relay driver circuit which is enabled by one of the eight Digital I/O Ports, C1 through C8 (see Section 14.9 for relay driver circuits). The 5 V ports can source up to 200 mA. An input protection transzorb will divert current to ground at approximately 10 V.

A functional description of the 37 pin connector located on the CR10 is provided in Appendix D.

14.9 SWITCHED 12 VOLT

A single switched 12 volt output is available for powering sensors or devices that require an unregulated 12 volts. The 12 volt output is limited to 600 mA of current.

A control port is used to operate the switched 12 volt control. Connect a wire from any control port to the switched 12 volt control (see Figure OV1.1-2 for location of ports). When the control port is set high, 12 volts is turned on to the switched 12 volt port. When the control port is set low, the switched 12 volts is turned off.

14.10 USE OF DIGITAL I/O PORTS FOR SWITCHING RELAYS

Each of the eight digital I/O ports can be configured as an output port and set low or high (0 V low, 5 V high) using I/O Instruction 20, Port Set, or commands 41 - 68 associated with Program Control Instructions 83 through 93. A digital output port is normally used to operate an external relay driver circuit because the port itself has a limited drive capability (1.5 mA at 3.5 V). Figure 14.10-1 shows a typical relay driver circuit in conjunction with a coil driven relay which may be used to switch external power to some device. In this example, when the control port is set high, 12 V from the datalogger passes through the relay coil, closing the relay which completes the power circuit to a fan, turning the fan on. Campbell

Scientific offers the A21REL-12 Four Channel Relay Driver (12 V coil) and the A6REL-12 Six Channel Relay Driver with manual override (12 V coil) for use with the CR10.

In other applications it may be desirable to simply switch power to a device without going through a relay. Figure 14.10-2 illustrates a circuit for switching external power to a device without going through a relay. If the peripheral to be powered draws in excess of 75 mA at room temperature (limit of the 2N2907A medium power transistor), the use of a relay (Figure 14.10-1) would be required.

Other control port activated circuits are possible for applications with greater current/voltage demands than shown in Figures 14.10-1 and 2. For more information contact Campbell Scientific's Marketing Department.

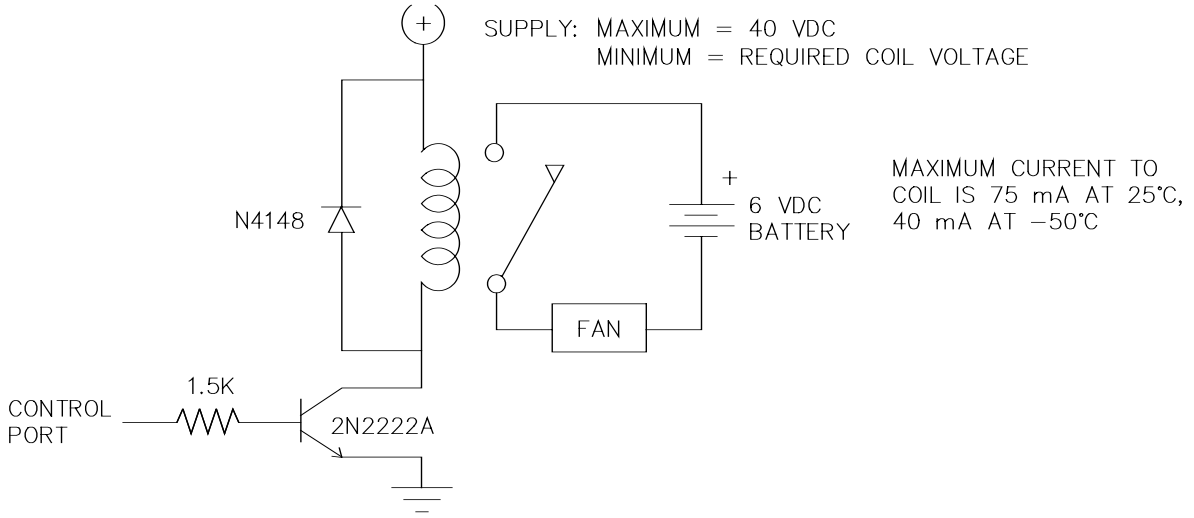


FIGURE 14.10-1. Relay Driver Circuit with Relay

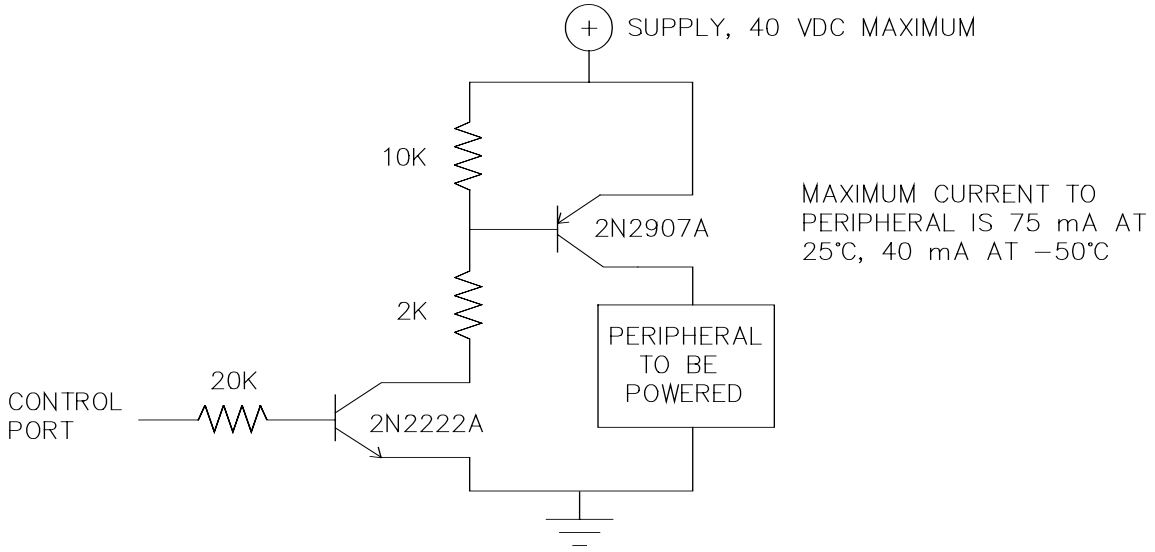


FIGURE 14.10-2. Power Switching without Relay

14.11 MAINTENANCE

The CR10 Wiring Panel and power supplies require a minimum of routine maintenance.

When not in use, the PS12LA should be stored in a cool, dry environment with the AC charging circuit activated.

The PS12ALK alkaline supply should not drop below 9.6 V before replacement. When not in use, remove the eight cells to eliminate potential corrosion of contact points and store in a cool dry place.

The CR10 module is sealed and contains desiccant to protect against the vagaries of humidity. The Wiring Panel and the connections between the Wiring Panel and the CR10 are still susceptible to humidity. To prevent corrosion at these points, additional desiccant must be placed inside the enclosure. If only alkaline batteries are inside the enclosure, the sensor lead entrance may be plugged to inhibit vapor transfer into the enclosure. Do not plug the entrance if lead acid batteries are present. Hydrogen gas generated by the batteries may build up to an explosive concentration.

APPENDIX A. GLOSSARY

ASCII: Abbreviation for American Standard Code for Information Interchange (pronounced "askee"). A specific binary code of 128 characters represented by 7 bit binary numbers.

ASYNCHRONOUS: The transmission of data between a transmitting and a receiving device occurs as a series of zeros and ones. For the data to be "read" correctly, the receiving device must begin reading at the proper point in the series. In asynchronous communication, this coordination is accomplished by having each character surrounded by one or more start and stop bits which designate the beginning and ending points of the information (see Synchronous).

BAUD RATE: The speed of transmission of information across a serial interface, expressed in units of bits per second. For example, 9600 baud refers to bits being transmitted (or received) from one piece of equipment to another at a rate of 9600 bits per second. Thus, a 7 bit ASCII character plus parity bit plus 1 stop bit (total 9 bits) would be transmitted in 9/9600 sec. = .94 ms or about 1000 characters/sec. When communicating via a serial interface, the baud rate settings of two pieces of equipment must match each other.

DATA POINT: A data value which is sent to Final Storage as the result of an Output Instruction. Strings of data points output at the same time make up Output Arrays.

EXECUTION INTERVAL: The time interval between initiating each execution of a given Program Table. If the Execution Interval is evenly divisible into 24 hours (86,400 seconds), the Execution Interval will be synchronized with 24 hour time, so that the table is executed at midnight and every execution interval thereafter. The table will be executed for the first time at the first occurrence of the Execution Interval after compilation. If the Execution Interval does not divide evenly into 24 hours, execution will start on the first even second after compilation. See Section OV4.3.1 for information on the choice of an Execution Interval.

EXECUTION TIME: The time required to execute an instruction or group of instructions. If the execution time of a Program Table exceeds the table's Execution Interval, the Program Table will be executed less frequently than programmed (Section OV4.3.1 and 8.9).

FINAL STORAGE: That portion of memory allocated for storing Output Arrays. Final Storage may be viewed as a ring memory, with the newest data being written over the oldest. Data in Final Storage may be displayed using the *7 Mode or sent to various peripherals (Sections 2, 3, and OV4.1).

GARBAGE: The refuse of the data communication world. When data are sent or received incorrectly (and there are numerous reasons this happens) a string of invalid, meaningless characters (garbage) results. Two common causes are: 1) a baud rate mismatch and 2) synchronous data being sent to an asynchronous device and vice versa.

HANDSHAKE, HANDSHAKING: The exchange of predetermined information between two devices to assure each that it is connected to the other. When not used as a clock line, the CLK/HS (pin 7) line in the CR10 is primarily used to detect the presence or absence of peripherals such as the Storage Module.

HIGH RESOLUTION: A high resolution data value has 5 significant digits and may range in magnitude from +.00001 to +99999. A high resolution data value requires 2 Final Storage locations (4 bytes). All Input and Intermediate Storage locations are high resolution. Output to Final Storage defaults to low resolution; high resolution output must be specified by Instruction 78.

INDEXED INPUT LOCATION: An Input location entered as an instruction parameter may be indexed by keying "C" before it is entered by keying "A"; two dashes (--) will appear at the right of the display. Within a loop (Instruction 87, Section 12), this will cause the location to be incremented with each pass through the loop. Indexing is also used with Instruction 75 to cause an Input location, which

normally remains constant, to be incremented with each repetition.

INPUT STORAGE: That portion of memory allocated for the storage of results of Input and Processing Instructions. The values in Input Storage can be displayed and altered in the *6 Mode.

INPUT/OUTPUT INSTRUCTIONS: Used to initiate measurements and store the results in Input Storage or to set or read Control/Logic Ports.

INSTRUCTION LOCATION NUMBER: As instructions are entered in a Program Table, they are numbered sequentially. The instruction location number gives the location of that instruction in the program sequence. When programming a table, the instruction location number and a P (e.g., 04: P00) prompts the user when it is time to enter an instruction.

INTERMEDIATE STORAGE: That portion of memory allocated for the storage of results of intermediate calculations necessary for operations such as averages or standard deviations. Intermediate storage is not accessible to the user.

LOOP: In a program, a series of instructions which are repeated a prescribed number of times, followed by an "end" instruction which exists the program from the loop.

LOOP COUNTER: Increments by 1 with each pass through a loop.

LOW RESOLUTION: The default output resolution. A low resolution data value has 4 significant decimal digits and may range in magnitude from +0.001 to +6999. A low resolution data value requires 1 Final Storage location (2 bytes).

MANUALLY INITIATED: Initiated by the user, usually with a keyboard, as opposed to occurring under program control.

MODEM/TERMINAL: Any device which: 1) has the ability to raise the CR10's ring line or be used with the SC32A to raise the ring line and put the CR10 in the Telecommunications Command State and 2) has an asynchronous serial communication port which can be configured to communicate with the CR10.

ON-LINE DATA TRANSFER: Routine transfer of data to a peripheral left on-site. Transfer is controlled by the program entered in the datalogger.

OUTPUT ARRAY: A string of data points output to Final Storage. Output occurs only when the Output Flag (Flag 0) is set. The first point of an Output Array is the Output Array ID, which gives the Program Table Number and the Instruction Location Number of the Instruction which sets the Output Flag. The data points which complete the Array are the result of the Output Processing Instructions which are executed while the Output Flag is set. The Array ends when the Output Flag is reset at the end of the table or when another Instruction acts upon the Output Flag. Output occurs only when the output flag is set. (Section 2.1)

OUTPUT INTERVAL: The time interval between initiations of a particular Output Array. Output occurs only when the Output Flag is set. The flag may be set at fixed intervals or in response to certain conditions (Sections OV4 and 1.2.1).

OUTPUT PROCESSING INSTRUCTIONS: Process data values and generate Output Arrays. Examples of Output Processing Instructions include Totalize, Maximize, Minimize, Average, etc. The data sources for these Instructions are values in Input Storage. The results of intermediate calculations are stored in Intermediate Storage. The ultimate destination of data generated by Output Processing Instructions is usually Final Storage but may be Input Storage for further processing. The transfer of processed summaries to Final Storage takes place when the Output Flag has been set by a Program Control Instruction.

PARAMETER: Used in conjunction with CR10 Program Instructions, parameters are numbers or codes which are entered to specify exactly what a given instruction is to do. Once the instruction number has been entered in a Program Table, the CR10 will prompt for the parameters by displaying the parameter number in the ID Field of the display.

PRINT DEVICE: Any device capable of receiving output over pin 6 (the PE line) in a receive-only mode. Printers, "dumb" terminals,

and computers in a terminal mode fall in this category.

PRINT PERIPHERAL: See Print Device.

PROCESSING INSTRUCTIONS: These Instructions allow the user to further process input data values and return the result to Input Storage where it can be accessed for output processing. Arithmetic and transcendental functions are included in these Instructions.

PROGRAM CONTROL INSTRUCTIONS: Used to modify the sequence of execution of Instructions contained in Program Tables; also used to set or clear flags.

PROGRAM TABLE: That portion of memory allocated for storing programs consisting of a sequence of user instructions which control data acquisition, processing, and output to Final Storage. Programming can be separated into 2 tables, each having its own user-entered Execution Interval. A third table is available for programming subroutines which may be called by instructions in Tables 1 or 2. The *1 and *2 Modes are used to access Tables 1 and 2. The *3 Mode is used to access Subroutine Table 3. The length of the tables is constrained only by the total memory available for programming (Section 1.5). Tables 1 and 2 have independent execution intervals. Table 1 execution has the higher priority; it may interrupt Table 2.

RING LINE (PIN 3): Line pulled high by an external device to "awaken" the CR10.

SAMPLE RATE: The rate at which measurements are made. The measurement sample rate is primarily of interest when considering the effect of time skew (i.e., how close in time are a series of measurements). The maximum sample rates are the rates at which measurements are made when initiated by a single instruction with multiple repetitions.

SIGNATURE: A number which is a function of the data and the sequence of data in memory. It is derived using an algorithm which assures a 99.998% probability that if either the data or its sequence changes, the signature changes.

SYNCHRONOUS: The transmission of data between a transmitting and receiving device occurs as a series of zeros and ones. For the data to be "read" correctly, the receiving device must begin reading at the proper point in the series. In synchronous communication, this coordination is accomplished by synchronizing the transmitting and receiving devices to a common clock signal (see Asynchronous).

THROUGHPUT: The throughput rate is the rate at which a measurement can be made, scaled to engineering units, and the reading stored in Final Storage. The CR10 has the ability to scan sensors at a rate exceeding the throughput rate (see SAMPLE RATE). The primary factor affecting throughput rate is the amount of processing specified by the user. In normal operation, all processing called for by an instruction must be completed before moving on the next instruction. The maximum throughput rate for a fast single-ended measurement is approximately 192 measurements per second (12 measurements, repeated 16 times per second). This rate is possible if the CR10's self-calibration function is suspended (this is accomplished by entering Instruction 24 into Program Table 2 while leaving the Execution Interval 0 so Program Table 2 never executes).

When the self-calibration function is operating, the maximum throughput rate for a fast, single-ended measurement is 192 measurements per second (12 measurements, 16 times per second).

APPENDIX B. CR10 PROM SIGNATURE AND OPTIONAL SOFTWARE

B.1 PROM SIGNATURE AND VERSION

The CR10 PROM signature is viewed by entering the *B Mode and advancing to window 2 (Section 1.6). The version number is in window 6 and the revision number in window 7.

The signatures of current standard PROMs are listed in Table B-1. If the CR10 has a Library Option PROM, the signature is listed on the calibration sheet shipped with the datalogger. The instructions for using the special features, can be found in Appendix H.

TABLE B-1. CR10 PROM Signatures

PROM DESCRIPTION	SIGNATURE *B, WINDOW 2	PROM NUMBER *B, WINDOW 6	REVISION NUMBER WINDOW 7
OS10-0.1	51903	.10000	0001
OS10-1.1	55030	1.1000	0001
OS10-0.1-4K	19464	90.100	0000

TABLE B-2. CR10 Library Options

INST. NO.	DESCRIPTION	BYTES
	Core files must be included	2578
13,14	Add R, S, & B Thermocouple Linearizations	456
15	Control Port Serial I/O	1111
23[1]	Burst Measurement	1588
27,28[1]	Period, Vibrating Wire Measurement	932
64	Paroscientific "T" Series Processing	169
101[1]	SDM-INT8	549
102[1]	SDM-SW8A	268
103,104[1]	SDM-AO4, SDM-CD16	239
SDM Utilities	216	
105,106	SDI12 Recorder/Sensor	1528
60	FFT	2297
62	Covariance/Correlation	796
81	Rainflow histogram	1159
97[1]	Initiate telecommunications	880
98[1]	Send Character	73
99	Satellite Interface	3143
64	Hz Pulse counters reset	-3
*4	Parameter Entry Table	575
*D	Add Tape up/down load	1196

[1] These instructions are in OS10-0.1 and are described in the standard manual.

B.2 AVAILABLE PROMS / LIBRARY OPTIONS

The set of instructions available in the CR10 is determined by the PROM (Programmable Read Only Memory that it is equipped with. If the standard PROM, OS10-0.1, does not fit the customer's application it is possible to create a library option PROM with the instructions and features desired, provided the memory capacity of the PROM is not exceeded.

Library Option PROMS are assembled from a base set of instructions plus selected functions in Table B.2-1. Consult a Campbell Scientific applications engineer for pricing and help in configuring a library option.

B.3 DESCRIPTION OF LIBRARY OPTIONS NOT IN STANDARD MANUAL

The following is a brief description of library options not found in the standard manual. If the

CR10 PROM contains one of the following options then detailed information on the special option(s) will be placed in Appendix H.

13,14 ADD R, S, & B THERMOCOUPLE LINEARIZATIONS
In addition to the linearizations for the T, E, J, and K thermocouples, Instructions 13 and 14 have the R, S, and B thermocouple linearizations.

15 CONTROL PORT SERIAL I/O (for smart sensor)
This Instruction allows sending and receiving half duplex data through the control ports at 300 or 1200 baud.

105,106 SDI12 RECORDER/SENSOR
Instruction 105 allows the CR10 to use its control ports to collect data from an intelligent sensor which uses the SDI12 communication protocol.

Instruction 106 allows the CR10 to behave as an SDI12 sensor.

62 COVARIANCE/CORRELATION
This Instruction calculates 1) means, 2) variances, 3) standard deviations, 4) covariances and 5) correlations for a set of input values and stores the results in Input Storage. Subintervals are allowed for convenient high pass filtering.

81 RAINFLOW HISTOGRAM
The Rainflow counting algorithm is a means of estimating cumulative damage fatigue by processing strain measurements to produce a histogram in which closed stress/strain hysteresis loops are counted by amplitude ranges. The histogram can be two dimensional, with each amplitude range sorted on the basis of average strain during the cycle. (The name, Rainflow, comes from the idea that a strain/time plot with strain as the y axis looks like a pagoda roof. Then, the algorithm is somehow analogous to the way in which rain flows down the roof.)

The Instruction can either process a swath of data from measurements made with the burst measurement instruction, or it can be used "on line", processing a new reading with each execution interval.

99 SATELLITE TRANSMITTER INTERFACE
Interface to allow output to GOES satellite via the Valcom satellite transmitter.

64 Hz RESET ON PULSE COUNTERS
The reset interval on the pulse counters is changed from 1/8 second to 1/64 second. This allows reading the pulse counters on intervals shorter than 0.125 second and allows counting frequencies up to 16 kHz with the 8 bit option. The quiescent current drain of the CR10 is increased to 5 mA when the pulse count instruction is used. A hardware modification to the CR10 is also required.

***4 PARAMETER ENTRY TABLE**
This option allows instruction parameters to be flagged. The values to use for the parameters are then entered into a table in the *4 Mode. This feature is of use when the same program is used for a number of stations but individual changes need to be made for each station (e.g. identical measurements are made at each station but certain sensors do not have interchangeable calibrations). It is easier and safer to train someone to enter the correct values in the table format than it would be to have them attempt to alter the program.

APPENDIX C. BINARY TELECOMMUNICATIONS

C.1 TELECOMMUNICATIONS COMMAND WITH BINARY RESPONSES

Command	Description
[no. of loc.]**F**	BINARY DUMP - CR10 sends, in Final Storage Format (binary, the number of Final Storage locations specified (from current MPTR locations), then Signature (no prompt).

DATALOGGER J AND K COMMANDS

3142J The 3142J command is used to toggle datalogger user flags, request final storage data, and to establish the input locations returned by the K command. The format of the command is as follows:

3142J<CR>abcd...nNULL

where
1) "3142J<CR>" is the command.

2) "a" is a 1 byte value representing the user flags to be toggled. The most significant bit (MSB), if set, will toggle datalogger user flag 8. Likewise, the 2nd most significant bit, if set, will toggle user flag 7, and so on to the least significant bit which, if set, toggles user flag 1. Toggle means that if a flag is set, it will be then reset, or if it is reset, it will be set.

3) "b" is a 1 byte value whose MSB will determine whether Final Storage Data is returned after the K command. If the bit is set, Final Storage Data, if any, will be returned after the next K command. The datalogger initially has this bit reset upon entering telecommunications, but once set by a J command, it will remain set until reset by another J command or telecommunications is terminated.

The 2nd MSB set means a port toggle byte will follow and port status is to be returned with the K command. Like the MSB, this bit is reset upon entering telecommunications, but remains set once set until reset by another J command or telecommunications is terminated. Currently only the CR10 datalogger recognizes this bit.

The remaining bits are reserved.

4) If the 2nd MSB in "b" was set then "c" is a port toggle byte, otherwise "c,d,...,n" are each 1 byte binary values each representing a datalogger input storage location. The data at those locations will be returned after the next K command. ASCII code 1 (0000001 binary) represents input location 1. ASCII codes 2 (00000010 binary) represents input location 2, and so on. The order of the location requests is not important. The list is limited, however, to 62 total location requests.

5) "Null" or ASCII code 0 (00000000 binary) terminates the J command. Alternately, 11111111 binary aborts the J command. If aborted, flags will not be toggled and location requests will not be saved.

User Enters	Datalogger Echo
3	3
1	1
4	4
2	2
J	J
CR	CR
	LF
	<
a	a
b	b
c	c
d	d
n	n
Null	Null

K The K command returns datalogger time, user flag status, port status if requested, the data at the input locations requested in the J command, and Final Storage Data if requested by the J command. The format of the command is K<CR> (K Return). The datalogger will echo the K and Return and send a Line Feed. The amount of data that follows depends on the J command previously executed; four time bytes, a user flags byte, four bytes for each input location requested in the J command, Final Storage data in CSI's binary format if requested by the J command, and terminating in 7F 00 HEX and two signature bytes.

User Enters	Datalogger Echo
K	K
CR	CR
	LF
	Time Minutes byte 1
	Time Minutes byte 2
	Time Tenths byte 1
	Time Tenths byte 2
	Flags byte
	Ports byte (if requested)
	Data1 byte 1
	Data1 byte 2
	Data1 byte 3
	Data1 byte 4
	Data2 byte 1
	Data2 byte 2
	Data2 byte 3
	Data2 byte 4
	DataN byte 1
	DataN byte 2
	DataN byte 3
	DataN byte 4
	Final Storage Data bytes
	01111111 binary byte
	00000000 binary byte
	Signature byte 1
	Signature byte 2

Time Minutes byte 1 is most significant. Convert from binary to decimal. Divide by 60 to get hours, the remainder is minutes. For example, 00000001 01011001 (01 59 HEX) is 345 decimal minutes or 5:45.

Time Tenths byte 1 is most significant. Convert from binary to decimal. Divide by 10 to get seconds and tenths of seconds. For example, 00000001 11000110 (01 C6 HEX) is 454 decimal or 45.4 seconds. Thus the datalogger time for 01 59 01 C6 HEX is 5:45:45.4.

The Flags byte expresses datalogger user flag status. The most significant bit represents Flag 8, and so on to the least significant bit which represents Flag 1. If a bit is set, the user flag is set in the datalogger.

The optional ports byte (currently on return if requested by a CR10 J command) expresses the datalogger port status. The most significant bit represents Port 8, and so on to the least significant bit which represents Port 1.

For each input location requested by the J command, four bytes of data are returned. The bytes are coded in Campbell Scientific, Inc. Floating Point Format. The format is decoded to the following:

$$\text{Sign}(\text{Mantissa}*2^{(\text{Exponent})})$$

Data byte 1 contains the Sign and the Exponent. The most significant bit represents the Sign; if zero, the Sign is positive, if one, the Sign is negative. The signed exponent is obtained by subtracting 40 HEX (or 64 decimal) from the 7 remaining least significant bits.

Data bytes 2 to 4 are a binary representation of the mantissa with byte 2 the most significant and 4 the least. The mantissa ranges in value from 80 00 00 hex (0.5 decimal) to FF FF FF HEX ($1-2^{-24}$ decimal). The one exception is for zero which is 00 00 00 00 HEX.

The Mantissa is calculated by converting Data bytes 2 to 4 into binary. Each bit represents some fractional value which is summed for all 24 bits. The bits are arranged from MSB to LSB with the most significant as bit_{23} and least significant as bit_0. The value that each bit represents = 2^{n-24}; where n=bit location. For example, if there was a 1 at bit_{20}, it's value would be $2^{(20-24)}$ or 2^{-4}.

A simple method for programming this is as follows:

Set Mantissa = 0.

Set Bit Value = 0.5.

For loop count = 1 to 24 do the following:

> If the MSB is one, then add Bit Value to the Mantissa.
>
> Shift the 24 bit binary value obtained from Data bytes 2 to 4 one bit to the left.
>
> Multiply Bit Value by 0.5.

End of loop.

Another method that can be used as an estimate is to convert Data bytes 2 to 4 from a long integer to floating point and dividing this value by 16777216.

As an example of a negative value, the datalogger returns BF 82 0C 49 HEX.

Data byte 1 = BF HEX.

Data byte 2 to 4 = 82 0C 49 HEX (or 8522825 decimal).

Data byte 1 is converted to binary to find the Sign. BF HEX = 10111111 BINARY.

The most significant bit is 1 so the Sign is NEGATIVE.

The exponent is found by subtracting 40 HEX from the remaining least significant bits. Converting the binary to hexadecimal, 111111 BINARY = 3F HEX (or 63 decimal).

3F - 40 HEX = FF FF FF FF FF HEX. Or in decimal: 63 - 64 = -1.

Exponent is -1 decimal.

The binary representation of Data bytes 2 to 4 is: 10000010 00001100 01001001.

Summing all the fractional values: $2^{-1} + 2^{-7} + 2^{-13} + 2^{-14} + 2^{-18} + 2^{-21} + 2^{-24} = 0.50800$.

Using the estimate method to find the Mantissa = 82 0C 49 HEX / 1 00 00 00 HEX (or 8522825 / 16777216) which is 0.50800 decimal.

The value is then $(-)0.508*2^{-1}$ which equals -0.254.

As an example of a positive value, the datalogger returns 44 D9 99 9A HEX.

Data byte 1 = 44 HEX.

Data byte 2 to 4 = D9 99 9A HEX (or 891290 decimal).

Data byte 1 is converted to binary to find the Sign. 44 HEX = 01000100 BINARY.

NOTE: Don't lose the leading zero!

The most significant bit is 0 so the Sign is POSITIVE.

The exponent is found by subtracting 40 HEX from the remaining least significant bits. Converting the binary to hexadecimal, 1000100 BINARY = 44 HEX (or 68 decimal).

44 - 40 HEX = 4 HEX. Or in decimal: 68 - 64 = 4.

Exponent is 4 decimal.

The binary equivalent of Data bytes 2 to 4 is: 11011001 10011001 10011010.

Summing all the fractional values:

$2^{-1} + 2^{-2} + 2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} + 2^{-12} + 2^{-13} + 2^{-16} + 2^{-17} + 2^{-20} + 2^{-21} + 2^{-23} = 0.85000$.

Using the estimate method to find the Mantissa = D9 99 9A HEX / 1 00 00 00 HEX (or 14260634 / 16777216) which is 0.85000 decimal.

The value is then $(+)0.85*2^{4}$ which equals 13.60.

If appropriately requested by a J command, Final Storage data, if any, will immediately follow the input location data. Refer to the datalogger manual for a description of how to decode Final Storage data in CSI's binary data format. Final Storage data will be limited to not more than 1024 bytes per K command.

The K command data is terminated with 7F 00 HEX (a unique binary format code) followed by two signature bytes. Refer to the datalogger manual for the meaning and calculation of the signature bytes. The signature in this case is a function of the first time byte through the 7F 00 HEX bytes. Calculate the signature of the bytes received and compare with the signature received to determine the validity of the transmission.

C.2 FINAL STORAGE FORMAT

CR10 data is formatted as either 2 byte LO Resolution or 4 byte HI Resolution values. The first two bytes of an Output Array contain a unique code (FC Hex) noting the start of the Output Array and the Output Array ID, followed by the 2 or 4 byte data values. At the end of the data sent in response to the telecommunications F command a 2 byte signature is sent (see below).

Representing the bits in the first byte of each two byte pair as ABCD EFGH (A is the most significant bit, MSB), the byte pairs are described here.

LO RESOLUTION FORMAT - D,E,F, NOT ALL ONES

Bits	Description
A	Polarity, 0 = +, 1 = -.
B, C	Decimal locators as defined below.
D-H plus second byte	13 bit binary value (D=MSB). Largest possible number without D, E, and F all 1 is 7167, but CSI defines the largest allowable range as 6999.

The decimal locators can be viewed as a negative base 10 exponent with decimal locations as follows:

B	C	Decimal Location
0	0	XXXX.
0	1	XXX.X
1	0	XX.XX
1	1	X.XXX

DATA TYPE WHEN D,E,F, ALL EQUAL ONE

If D, E, and F are all ones, the data type is determined by the other bits as shown below. X implies a "don't care" condition; i.e., the bit can be either 1 or 0 and is not used in the decode decision.

A	B	C	D	E	F	G	H	DATA TYPE AND SECOND BYTE FORMAT
1	1	1	1	1	1	X	X	A,B,C, = 1 - Start of Output Array, G & H are the most significant bits of the Output Array ID. All 8 bits of the 2nd byte are also included in the ID.
X	X	0	1	1	1	X	X	C = 0 - First byte of a 4 byte value.
0	0	1	1	1	1	X	X	A,B = 0; C = 1 - Third byte of a 4 byte value.
0	1	1	1	1	1	1	1	A = 0; remaining bits = 1 - First byte of a 2 byte "dummy" word. The CR10 always transmits a 0 for the 2nd byte, but the word can be decoded on the basis of the 1st byte only.

HI RESOLUTION FORMAT

Continuing to use the A-H bit representation, the four byte number is shown below as two two byte pairs.

AB0111GH XXXXXXXX 001111GH XXXXXXXX

BITS, 1ST BYTE, 1ST PAIR	DESCRIPTION
CDEF = 0111	Code designating 1st byte pair of four byte number.
B	Polarity , 0 = +, 1 = -.
G,H,A,	Decimal locator as defined below.
2nd byte	16th - 9th bit (left to right) of 17 bit binary value.
ABCDEF = 001111	Code designating 2nd byte pair of four byte number.
G	Unused bit.
H	17th and MSB of 17 bit binary value.
2nd byte	8th - 1st bit (left to right) of 17 bit binary value.

CSI defines the largest allowable range of a high resolution number to be 99999.

Interpretation of the decimal locator for a 4 byte data value is given below. The decimal equivalent of bits GH is the negative exponent to the base 10.

BITS G H A	DECIMAL FORMAT 5 digits
0 0 0	XXXXX.
0 0 1	XXXX.X
0 1 0	XXX.XX
0 1 1	XX.XXX
1 0 0	X.XXXX
1 0 1	.XXXXX

C.3 GENERATION OF SIGNATURE

At the end of a binary transmission, a signature is sent. The signature is a 2 byte integer value which is a function of the data and the sequence of data in the Output Array. It is derived with an algorithm that assures a 99.998% probability of detecting a change in the data or its sequence. The CR10 calculates the signature using each transmitted byte beginning with the Final Storage format data (for K command, echo and carriage return line feed are not included) until the 2 byte signature itself. By calculating the signature of the received data and comparing it to the transmitted signature, it can be determined whether the data was received correctly.

SIGNATURE ALGORITHM

- S1,S0 - represent the high and low bytes of the signature, respectively
- M - represents a transmitted data byte
- n - represents the existing byte
- n+1 - represents the new byte
- T - represents a temporary location
- C - represents the carry bit from a shift operation

1. The signature is initialized with both bytes set to hexadecimal AA.

$$S_1(n) = S_0(n) = AA$$

2. When a transmitted byte, M(n+1), is received, form a new highsignature byte by setting it equal to the existing low byte. Save the old high byte for later use.

$$T_1 = S_1(n)$$

$$S_1(n+1) = S_0(n)$$

3. Form a temporary byte by shifting the old low signature byte one bit to the left and adding any carry bit which results from the shift operation. A "shift left" is identical to a multiply by 2. Ignore any carry bit resulting from the add.

$$T_2 = \text{shift left } (S_0(n)) + \text{carry}$$

4. Form the new low signature byte by adding the results of operation 3 to the old high signature byte and the transmitted byte. Ignore any carry bits resulting from these add operations.

$$S_0(n+1) = T_2 + S_1(n) + M(n+1)$$

As each new transmitted byte is received, the procedure is repeated.

APPENDIX D. CR10 37 PIN PORT DESCRIPTION

PIN #	DESCRIPTION	PIN #	DESCRIPTION
1	12V	19	C1
2	6L	20	G
3	AG	21	6H
4	5H	22	5L
5	4L	23	AG
6	AG	24	4H
7	3H	25	3L
8	2L	26	AG
9	AG	27	2H
10	1H	28	1L
11	EX CTRL 3	29	AG
12	EX CTRL 2	30	E3
13	EX CTRL 1	31	E2
14	AG	32	E1
15	P1	33	P2
16	C7	34	C8
17	C5	35	C6
18	C3	36	C4
		37	C2

APPENDIX E. ASCII TABLE

American Standard Code for Information Interchange
Decimal Values and Characters

(X3.4-1968)

Dec.	Char.	Dec.	Char.	Dec.	Char.	Dec.	Char.
0	CONTROL @	32	SPACE	64	@	96	`
1	CONTROL A	33	!	65	A	97	a
2	CONTROL B	34	"	66	B	98	b
3	CONTROL C	35	#	67	C	99	c
4	CONTROL D	36	$	68	D	100	d
5	CONTROL E	37	%	69	E	101	e
6	CONTROL F	38	&	70	F	102	f
7	CONTROL G	39	'	71	G	103	g
8	CONTROL H	40	(	72	H	104	h
9	CONTROL I	41	)	73	I	105	i
10	CONTROL J	42	*	74	J	106	j
11	CONTROL K	43	+	75	K	107	k
12	CONTROL L	44	,	76	L	108	l
13	CONTROL M	45	-	77	M	109	m
14	CONTROL N	46	.	78	N	110	n
15	CONTROL O	47	/	79	O	111	o
16	CONTROL P	48	0	80	P	112	p
17	CONTROL Q	49	1	81	Q	113	q
18	CONTROL R	50	2	82	R	114	r
19	CONTROL S	51	3	83	S	115	s
20	CONTROL T	52	4	84	T	116	t
21	CONTROL U	53	5	85	U	117	u
22	CONTROL V	54	6	86	V	118	v
23	CONTROL W	55	7	87	W	119	w
24	CONTROL X	56	8	88	X	120	x
25	CONTROL Y	57	9	89	Y	121	y
26	CONTROL Z	58	:	90	Z	122	z
27	CONTROL [	59	;	91	[	123	{
28	CONTROL \	60	<	92	\	124	|
29	CONTROL]	61	=	93	]	125	}
30	CONTROL ^	62	>	94	^	126	~
31	CONTROL _	63	?	95	_	127	DEL

APPENDIX G. CHANGING RAM OR PROM CHIPS

The CR10 has two sockets for Random Access Memory (RAM) and one socket for Programmable Read Only Memory (PROM). The standard CR10 has 64K of RAM, (a 32K RAM chip in each socket). Earlier CR10s had 16K of RAM (an 8K RAM chip in each socket).

G.1 DISASSEMBLING THE CR10

The sockets provided for RAM and PROM are located on the CR10 CPU circuit card inside the CR10 can. To expose the RAM and PROM sockets, remove the two phillips head screws from the end opposite the connectors. Remove the end cap. The ends of two circuit cards and the RF shield will be visible (see Figure G-1). Now lay the CR10 on a flat surface, (i.e., a table), and push on the RF shield with your thumbs while grasping the can with your hands. Remove the circuit cards from the can. Orient the cards with the connector on the left and with the card that matches Figure G-2. The Central Processing Unit (CPU) is found at location H-9 and the three slots for RAM and PROM will be directly beneath it.

G.2 INSTALLING NEW RAM CHIPS IN CR10s WITH 16K RAM

The two 8K RAM chips are found at locations C-11 and C-14. With a small flat screw driver gently pry out the two 8K RAM chips at these locations and replace them with the 32K RAM chips provided in the memory upgrade. The new chips should be installed so the notched end is towards the nearest card edge. Before pushing the chips into the socket make certain that all the pins are correctly seated. After installing the 32K chips check for pins that may be bent or not firmly seated in the socket. If you notice a bent pin, remove the chip, carefully straighten it and repeat the installation procedure.

G.2.1 CHANGING JUMPERS

There are six jumpers used to configure hardware for different RAM sizes. Figure 2 shows the jumper settings for different memory configurations. A pin or small screw driver tip will work best for pulling these jumpers and relocating them as shown in Figure 2.

G.2.2 RAM TEST

Attach the CR10KD Keyboard/Display and apply power to the CR10. After the CR10 executes the RAM/PROM self test, the number 96 should be displayed in the window. The number is the sum of Kbytes in RAM (64) plus the number of Kbytes in ROM (32).

G.3 INSTALLING NEW PROM

The PROM chip is found at location C8 on the CR10 CPU board, (see Figure G-2). With a small flat screw driver, gently pry out the PROM chip and replace it with the new one. The new chip should be installed so that the notched end is towards the nearest card edge. Before pushing the chip into the socket make certain that all the pins are seating correctly. After installing the chip check for pins that may be bent or not making contact. If you notice a bent pin, remove the chip, carefully straighten it and repeat the installation procedure.

To make certain that the new chip is installed correctly enter the CR10 *B mode, (Section 1.6), and advance to the second window. This window displays the PROM signature. The five digit number in the window should match the PROM signature given with the new PROM documentation. If the numbers are different disassemble the CR10 and look for pins that are bent or not firmly seated.

G.4 INSTALLING 4K PROGRAM MEMORY PROM

Newer CR10s (shipped after 11-4-93) can be converted from the standard 2K Program Memory to 4K Program Memory by installing the correct PROM and moving a jumper. Figure G-3 shows the location and settings for the jumper. Install the PROM as described in Section G.3.

Older CR10s do not have this jumper and must be sent to Campbell Scientific for a hardware modification to 4K Program Memory.

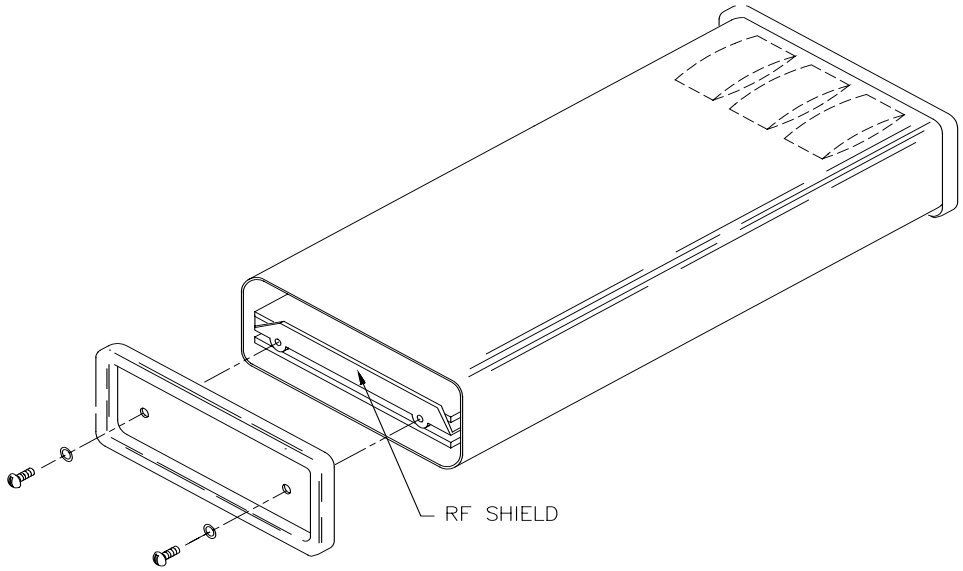


FIGURE G-1. Disassembling CR10

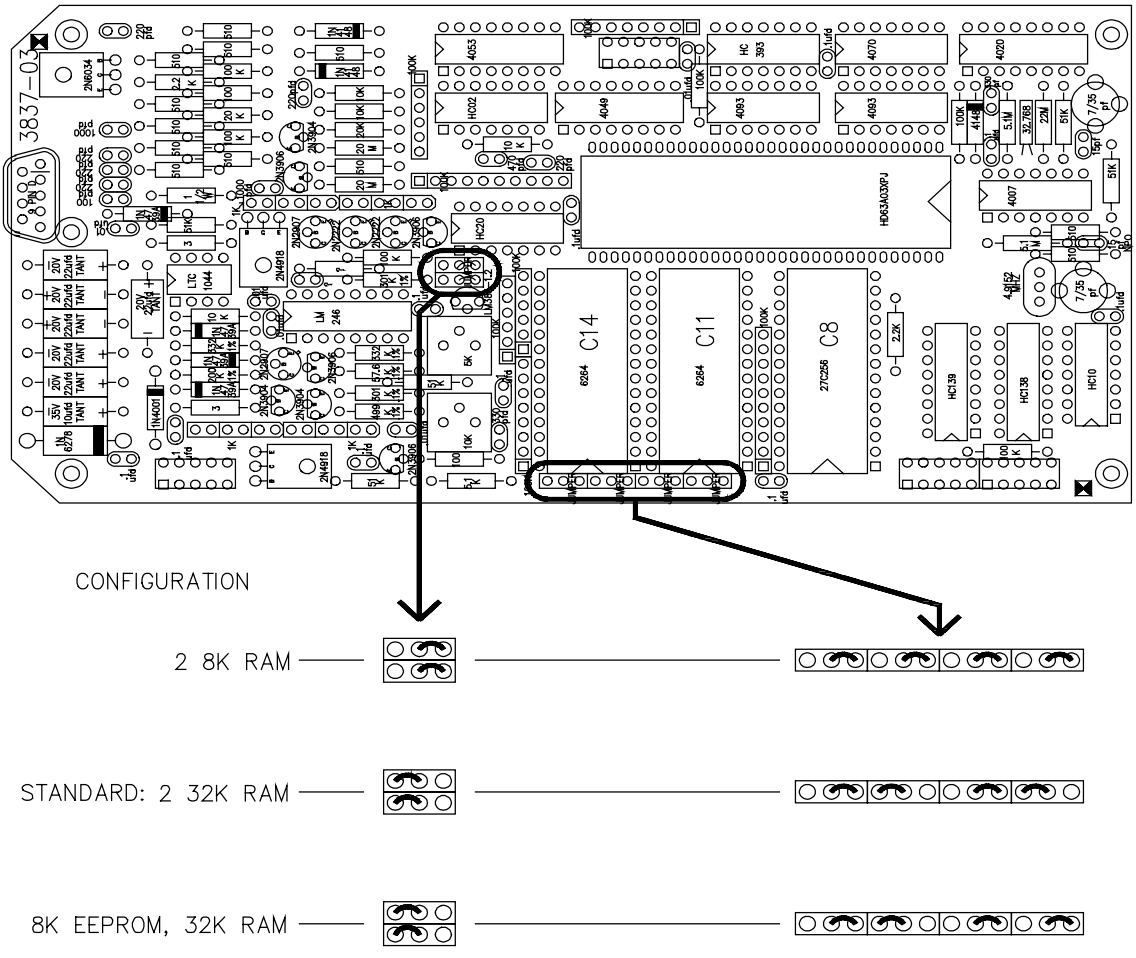


FIGURE G-2. Jumper Settings for Different RAM Configurations in Early CR10s

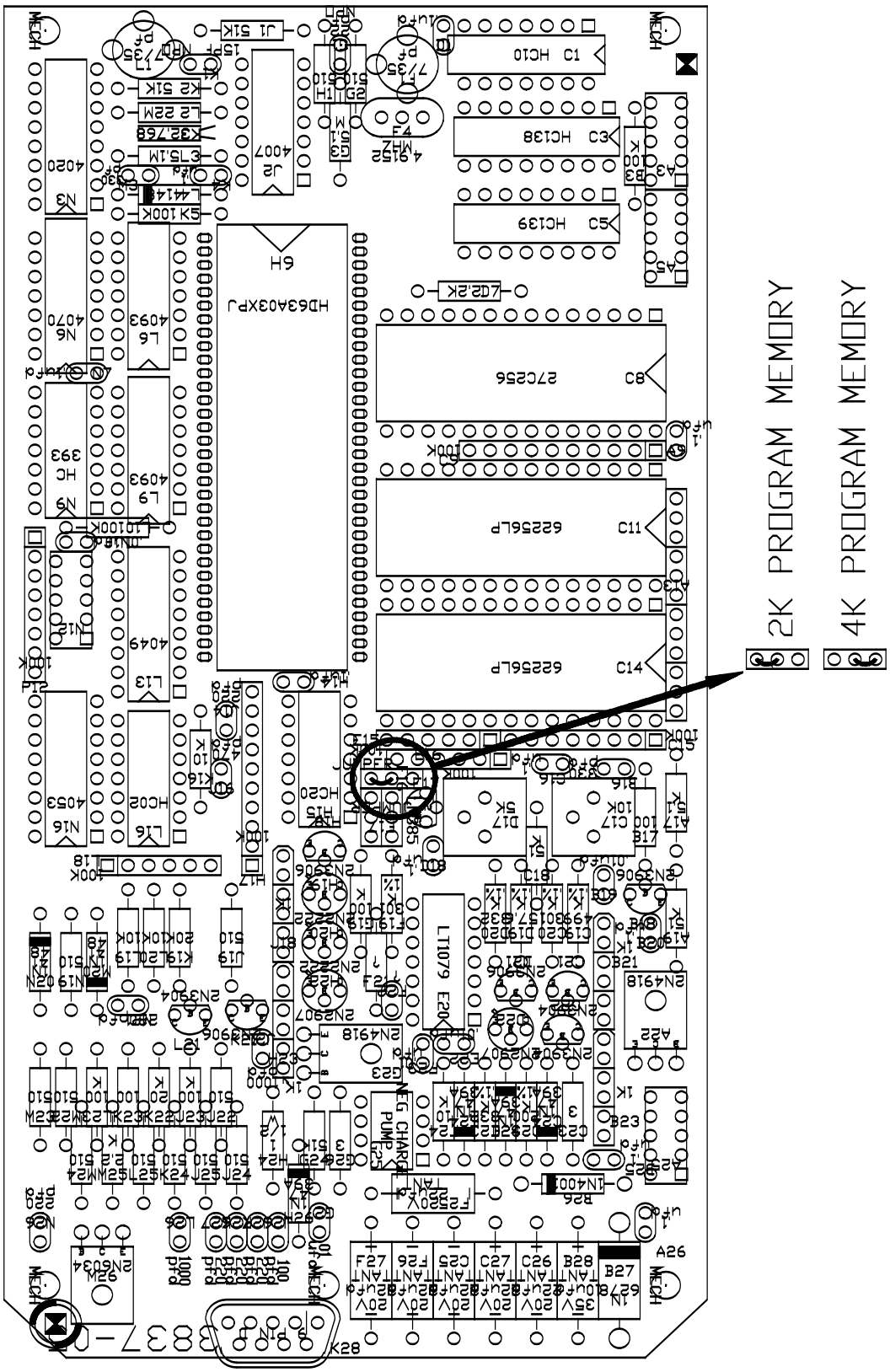


FIGURE G-3. Jumper Settings and Locations

LIST OF TABLES

PAGE

OVERVIEW

OV4.1-1 * Mode Summary OV-10
OV4.2-1 Key Definition/Editing Functions OV-10
OV4.2-2 Additional Keys Allowed In Telecommunications OV-11
OV6.1-1 Data Retrieval Methods and Related Instructions OV-18
OV6.1-2 Data Retrieval Sections in Manual OV-18

1. FUNCTIONAL MODES

1.2-1 Sequence of Time Parameters in *5 Mode 1-3
1.3-1 *6 Mode Commands 1-3
1.5-1 Memory Allocation in CR10 1-5
1.5-2 Description of *A Mode Data 1-5
1.6-1 Description of *B Mode Data 1-6
1.7-1 *C Mode Entries 1-7
1.8-1 *D Mode Commands 1-7
1.8-2 ASCII and Storage Module Command Options 1-8
1.8-3 Program Load Error Codes 1-8
1.8-4 Example Program Listing From *D Command 1 1-8

2. INTERNAL DATA STORAGE

2.2-1 Resolution Range Limits of CR10 Data 2-3
2.3-1 *7 Mode Command Summary 2-4

3. INSTRUCTION SET BASICS

3.5-1 Input Voltage Ranges and Codes 3-2
3.7-1 Flag Description 3-3
3.7-2 Example of the Use of Flag 9 3-4
3.8-1 Command Codes 3-4
3.9-1 Input/Output Instruction Memory and Execution Times 3-6
3.9-2 Processing Instruction Memory and Execution Times 3-7
3.9-3 Output Instruction Memory and Execution Times 3-7
3.9-4 Program Control Instruction Memory and Execution Times 3-8
3.10-1 Error Codes 3-9

4. EXTERNAL STORAGE PERIPHERALS

4.1-1 Output Device Codes for Instruction 96 and *8 Mode 4-2
4.2-1 *8 Mode Entries 4-3
4.3-1 Cassette Recorder Specifications 4-4
4.3-2 Format 2 Specifications 4-5
4.6-1 *9 Commands for Storage Module 4-8

PAGE

5. TELECOMMUNICATIONS

5.1-1 Telecommunications Commands 5-3

6. 9 PIN SERIAL INPUT/OUTPUT

6.1-1 Pin Description 6-1
6.6-1 SD Addresses 6-5
6.7-1 SC32A Pin Description 6-6
6.7-2 DTE Pin Configuration 6-6

7. MEASUREMENT PROGRAMMING EXAMPLES

7.16-1 Calibration Data for Sensor 3998 7-16
7.17-1 Period Averaging Instruction 27 7-19
7.17-2 Input Frequency Gain Codes 7-19
7.17-3 Coefficient Entry Format for Parosci. "T" Series Pressure Transducer Instruction 30 7-20

9. INPUT/OUTPUT INSTRUCTIONS

9-1 Input Voltage Ranges and Codes 9-1
9-2 Pulse Count Configuration Codes 9-3
9-3 Thermocouple Type Codes 9-7
9-4 Voltage and Temperature Ranges for Thermocouples If the Reference is 20°C 9-7
9-5 Port Configuration Option Codes 9-9
9-6 SDI-12 Command Codes 9-16

12. PROGRAM CONTROL INSTRUCTIONS

12-1 Flag Description 12-1
12-2 Command Codes 12-1
12-3 Loop Example: Block Data Transform 12-3
12-4 Example: Loop with Delay 12-4
12-5 Comparison Codes 12-4

13. CR10 MEASUREMENTS

13.3-1 Exponential Decay, Percent of Maximum Error vs. Time in Units of τ 13-4
13.3-2 Properties of Three Belden Lead Wires Used by Campbell Scientific 13-6
13.3-3 Settling Error, in Degrees, for 024A Wind Direction Sensor vs. Lead Length 13-7
13.3-4 Measured Peak Excitation Transients for 1000 Foot Lengths of Three Belden Lead Wires Used by Campbell Scientific 13-8
13.3-5 Summary of Input Settling Data for Campbell Scientific Resistive Sensors 13-9
13.3-6 Maximum Lead Length vs. Error for Campbell Scientific Resistive Sensors 13-9
13.3-7 Source Resistances and Signal Levels for YSI #44032 Thermistor Configurations Shown in Figure 13.3-7 (2V Excitation) 13-10
13.4-1 Limits of Error for Thermocouple Wire (Reference Junction at 0°C) 13-13
13.4-2 Limits of Error on CR10 Thermocouple Output Linearization 13-15
13.4-3 Ref. Temp. Compensation Range & Linearization Error Relative to NBS Standards 13-15
13.4-4 Example of Errors in Thermocouple Temperature 13-16
13.5-1 Comparison of Bridge Measurement Instructions 13-19
13.5-2 Calculating Resistance Values from Bridge Measurement 13-20

PAGE

14. INSTALLATION AND MAINTENANCE

14.2-1 Typical Current Drain for Common CR10 Peripherals 14-1
14.3-1 Typical Alkaline Battery Service and Temperature..... 14-3
14.3-2 PS12LA Battery and AC Transformer Specifications 14-4

APPENDIX B. CR10 PROM SIGNATURE AND OPTIONAL SOFTWARE

B-1 CR10 PROM Signature..... B-1
B-2 CR10 Library Options B-1

LIST OF FIGURES

PAGE

OVERVIEW

OV1.1-1 CR10 and Wiring Panel OV-2
OV1.1-2 CR10 Wiring Panel/Instruction Access OV-3
OV2.1-1 Instruction Types and Storage Areas OV-6
OV2.3-1 Program and Subroutine Tables OV-8
OV6.1-1 Data Retrieval Hardware Options OV-19

2. INTERNAL DATA STORAGE

2.1-1 Ring Memory Representation of Final Data Storage 2-1
2.1-2 Output Array ID 2-2

3. INSTRUCTION SET BASICS

3.8-1 If Then/Else Execution Sequence 3-4
3.8-2 Logical AND Construction 3-5
3.8-3 Logical OR Construction 3-5

4. EXTERNAL STORAGE PERIPHERALS

4.4-1 Example of CR10 Printable ASCII Output Format 4-6

6. 9 PIN SERIAL INPUT/OUTPUT

6.1-1 9 Pin Female Connector 6-1
6.2-1 Hardware Enabled and Synchronously Addressed Peripherals 6-2
6.3-1 Servicing of Ring Interrupts 6-3
6.6-1 Addressing Sequence for the RF Modem 6-4
6.7-1 Transmitting the ASCII Character 1 6-7

7. MEASUREMENT PROGRAMMING EXAMPLES

7.1-1 Wiring Diagram for LI200S 7-1
7.2-1 Typical Connection for Active Sensor with External Battery 7-2
7.3-1 CR10TCR Mounted on the CR10 Wiring Panel 7-3
7.4-1 Thermocouples with External Reference Junction 7-4
7.7-1 Wiring Diagram for Anemometer 7-5
7.8-1 Wiring Diagram for Rain Gage with Long Leads 7-6
7.9-1 Wiring Diagram for PRT in 4 Wire Half Bridge 7-7
7.10-1 3 Wire Half Bridge Used to Measure 100 ohm PRT 7-8
7.11-1 Full Bridge Schematic for 100 ohm PRT 7-9
7.12-1 Wiring Diagram for Full Bridge Pressure Transducer 7-10
7.13-1 Lysimeter Weighing Mechanism 7-11
7.13-2 6 Wire Full Bridge Connection for Load Cell 7-12
7.14-1 6 227 Gypsum Blocks Connected to the CR10 7-13
7.15-1 101 Thermistor Probes Connected to CR10 7-14
7.16-1 A Vibrating Wire Sensor 7-15
7.16-2 Well Monitoring Example 7-17
7.16-3 Hook up to AVW1 7-18
7.17-1 CR10/Paroscientific "T" Series Transducer Wiring Diagram 7-21

PAGE

8. PROCESSING AND PROGRAM CONTROL EXAMPLES

8.3-1 AM416 Wiring Diagram for Thermocouple and Soil Moisture Block Measurements 8-4
8.5-1 Connections for Rain Gage..... 8-6

9. INPUT/OUTPUT INSTRUCTIONS

9-1 Conditioning for Long Duration Voltage Pulses..... 9-2

13. CR10 MEASUREMENTS

13.1-1 50 and 60 Hz Noise Rejection 13-1
13.2-1 Timing of Single-Ended Measurement 13-2
13.2-2 Differential Voltage Measurement Sequence 13-2
13.3-1 Input Voltage Rise and Transient Decay 13-4
13.3-2 Typical Resistive Half Bridge 13-5
13.3-3 Source Resistance Model for Half Bridge Connected to the CR10..... 13-5
13.3-4 Wire Manufacturers Capacitance Specifications, Cw..... 13-6
13.3-5 Model 024A Wind Direction Sensor..... 13-6
13.3-6 Resistive Half Bridge Connected to Single-Ended CR10 Input..... 13-7
13.3-7 Half Bridge Configuration for YSI #44032 Thermistor Connected to CR10 13-11
13.3-8 Measuring Input Settling Error with the CR10 13-12
13.3-9 Incorrect Lead Wire Extension on Model 107 Temperature Sensor 13-12
13.4-1 Thermistor Polynomial Error 13-14
13.4-2 Diagram of Junction Box..... 13-16
13.5-1 Circuits Used with Instructions 4-9 13-18
13.5-2 Excitation and Measurement Sequence for 4 Wire Full Bridge..... 13-19
13.6-1 AC Excitation and Measurement Sequence for AC Half Bridge..... 13-21
13.6-2 Model of Resistive Sensor with Ground Loop 13-21

14. INSTALLATION AND MAINTENANCE

14.3-1 PS12 12 Volt Power Supply and Charging Regulator 14-3
14.6-1 Connecting to Vehicle Power Supply..... 14-5
14.7-1 Wiring Panel Grounding Diagram and Excitation Control..... 14-6
14.10-1 Relay Driver Circuit with Relay 14-8
14.10-2 Power Switching without Relay..... 14-8

APPENDIX G. CHANGING RAM OR PROM CHIPS

G-1 Disassembling CR10 .....G-2
G-2 Jumper Settings for Different RAM Configurations.....G-2
G-3 Jumper Settings and Locations G-3

CR10 INDEX

* Modes, See Modes
1/X - [Instruction 42] **10-2**
107 Thermistor Probe - [Instruction 11] **9-5**
  Programming examples **7-3**
CR10TCR Thermocouple Reference **7-3**
12V terminals **OV-3**, **OV-4**
100 ohm PRT
  3 wire half bridge **7-8**
  4 wire half bridge **7-6**
  4 wire full bridge **7-9**
207 Relative Humidity Probe - [Instruction 12] **9-6**
  Programming example **7-4**
227 Gypsum Soil Moisture Block **7-13**
3 Wire Half Bridge - [Instruction 7] **9-4**, **13-18**, **13-19**, **13-20**
  Example **7-8**
4 Wire Full Bridge - [Instruction 6] **9-4**, **13-18**, **13-19**, **13-20**
  Example **7-9**
5th Order Polynomial - [Instruction 55] **10-5**
  Programming examples **7-14**, **7-15**
5V outputs **OV-3**, **OV-4**, **14-5**

A

A*X + B, see Scaling Array with Multiplier and Offset - [Instruction 53]
A04 Analog Output, see SDM-A04
A6Rel-12/A21Rel-12 Relay Driver **14-7**
ABS(X) - [Instruction 43] **10-3**
AC Half Bridge - [Instruction 5] **9-3**
  Programming examples **7-14**, **8-4**
Activate Serial Data Output - [Instruction 96] **4-1**, **12-6**
  Interrupts during **6-3**
  Programming examples **OV-17**
  Using Storage Module with **4-7**
Alkaline power supply **14-2**
AM416 Multiplexer
  Measuring soil moisture blocks **8-3**
  Using control ports and loop **8-3**
Analog ground definition **OV-3**, **OV-4**
Analog inputs **OV-3**, **OV-4**
Analog output, see SDM-A04
Analog to Digital (A/D) conversion **13-1**
AND construction, Logical **3-5**
Anemometer (Photochopper output)
  Programming example **7-5**
Array ID
  and Output Interval **A-2**
  Calculating data points **4-6**
  Definition **A-2**
  Setting ID **2-1**, **11-5**
ASCII
  Characters **E-1**
  Definition **A-1**
  Output formats **4-6**
  Program transfer to computer/printer **1-8**
  Table **E-1**
  Terminal or computer with terminal emulator **OV-9**
  Table **E-1**
  Transmission **6-7**
Asynchronous, Definition **A-1**
Automatic calibration sequence **13-22**
Average - [Instruction 71] **11-3**
  Programming example **OV-16**
Average, Computing running **8-1**
AVW1/AVW4 Vibrating Wire Interface **7-15**

B

Battery Power **14-2**
Battery Voltage - [Instruction 10] **9-5**
Baud rate
  Definition **6-7**, **A-1**
Begin Case Statement - [Instruction 93] **12-5**
Binary telecommunications **C-1**
  Generation of signature **C-4**
Block Move - [Instruction 54] **10-4**
  Programming example **8-1**
Bridge measurements **13-17**
  3 Wire Half Bridge 100 ohm PRT **7-8**
  4 Wire Full Bridge (Pressure Transducer) **7-10**
  4 Wire Full Bridge 100 ohm PRT **7-9**
  4 Wire Half Bridge 100 ohm PRT **7-6**
  6 Wire Full Bridge (Lysimeter) **7-11**
Comparison of bridge measurement instructions **13-18**
Diagram of bridge measuring circuits **13-18**
  with AC excitation **13-19**
Bridge Transform - [Instruction 59] **10-6**
Programming examples **7-10**, **7-14**, **8-4**
Burst Measurement - [Instruction 23] **9-9**

C

C20 Cassette Interface **4-3**
Cables/Leads
  Avoid PVC insulated conductors **13-10**
  Connecting Leads to Wiring Panel **OV-1**

Effect of lead length on signal settling time **13-3**
Tipping bucket rain gage with long leads **7-6**
Calibration - [Instruction 24] **9-12**
Process **13-22**
Cassette recorder **4-4**
Cautionary notes **vii**
CD16, see SDM-CD16 Control Port Expansion Module
Channels
Differential analog **OV-3**, **13-2**
Single-ended analog **OV-3**, **13-2**
Checksum **5-2**
Clock
Example of setting **OV-17**
Set/display time (*5 Mode) **1-2**
CM6/CM10 Tripods grounding protection **14-6**
Common mode range **13-3**, **14-7**
Communicating with the CR10 **OV-8**
Via telemetry **5-1**
With external peripherals **4-1**
Communication
Protocol **OV-8**, **6-7**, **5-2**
Troubleshooting **6-7**
Compiling
Data **1-4**
Errors **3-9**
Program **1-2**
Computation of Running Average **8-1**
Computer
Connecting to serial port **OV-9**
Save/load program (*D Mode) **1-9**
Use with SC32A **OV-9**, **6-5**
Connecting Power to the CR10 **OV-3**, **OV-5**
Control Port Expansion Module with Drivers, see SDM-CD16
Control/logic ports
Command codes affecting **OV-3**, **12-1**
Configuration, see Port Set - [Instruction 20]
Definition **OV-4**
Display/toggle [Instruction 20] **1-4**, **9-8**, **C-1**
Indexing **3-1**
Voltages > 5.5, Applying **vii**
Cosine **10-3**
Counter
Pulse Count - [Instruction 3] **9-1**
Setting to 16 bit accumulator **9-2**
CR10 Pin Port description **D-1**
CR10KD Keyboard Display and SC12 cable **OV-9**
Displaying Stored Data (*7 Mode) **2-3**
Entering programs with **OV-12**
Key Definition **OV-10**
Typical Current Drain **14-1**
Wiring Panel
Connecting to vehicle power supply **14-5**
Description **OV-1, 14-7**

D

Data point
Definition **A-1**
Number per Output Array **4-6**
Data retrieval, External storage peripherals
General **4-1**
Hardware options **OV-17**
Manually initiated (*8 Mode) **4-3**
Methods and related instructions **OV-18**
On-line (Instruction 96) **4-1**, **12-6**
Print device **4-2**
Print formats **4-5**
Site Visitations, Estimating time between **4-3**
Storage Module **4-6**
Tape recorder **4-3**
Data Set Ready (DSR) **6-6**
Data Storage Pointer (DSP) **2-1**, **4-2**
Data Terminal Equipment (DTE) pin configuration **6-6**
Data transfer
ASCII vs. binary **5-1**
Interrupts During **6-3**
Manual (*8 Mode) **4-3**
On-line (Instruction 96) **4-1**, **12-6**
Data transmission, See Data retrieval
Data types parameter **3-1**
Dataloggers, Differences between CR10 and other **OV-1**
Date (*5 Mode), Setting/displaying **1-2**
DC103A/DC112 Phone Modem current drain, typical **14-1**
Desiccant
in Enclosure **14-9**
Replacement in CR10 **14-1**
Differential measurements on analog inputs **OV-4**, **13-2**
Differential Voltage w/ Excitation & Delay - [Instruction 8] **9-1**
Differential Volts - [Instruction 2] **9-2**
Programming example **7-2**
Direct battery connection **14-5**
Display Pointer (DPTR) **2-2**
Display, See CR10KD Keyboard/Display
Displaying/Altering Input Memory, Flags and Ports-*6 Mode **1-3**
Divide
X/Y - [Instruction 38] **10-2**
X Mod F - [Instruction 46] **10-3**
DO - [Instruction 86] **12-2**
Programming example **OV-13**
DPTR **2-2**
Druck PDCR 930 depth pressure transducer
Programming example **7-10**

DSP **2-1**
DSR (Data Set Ready) **6-6**
DTE (Data Terminal Equipment) pin configuration **6-6**
Duplex, Definition **6-7**

E

Earth Ground **OV-4**, **14-6**
Editing datalogger programs **OV-15**
Editor errors **3-8**
EDLOG **OV-12**, **5-4**
ELSE - [Instruction 94] **12-6**
Programming example **8-6**
Enclosures, Environmental **14-1**
END - [Instruction 95] **12-6**
Error codes **3-8**
Overranging **vi**, **3-2**
Overrun occurrences **1-1**
Examples, programming **OV-12**
Ex-Del-Diff - [Instruction 8] **9-4**
Ex-Del-SE - [Instruction 4] **9-3**
Excit-Del - [Instruction 22] **9-9**
Excitation outputs **OV-3**
Excitation with Delay - [Instruction 22] **9-9**
Programming example **8-6**
Excite, Delay, and Measure - [Instruction 4] **9-3**
Programming examples **7-15**
Execution interval **OV-6**, **1-1**, **A-1**, **OV-7**
Example of entering **OV-13**
Exceeding **1-1**
Execution time
Definition **A-1**
Program instruction **3-6**
Exp(X) - [Instruction 41] **10-2**
External power supply **14-2**
External Reference Junction **7-3**
External Storage Peripherals **4-1**
Cassette Tape Option **4-3**
Manually Initiated Data Output **4-3**

F

Fast and Slow Measurement Sequence **13-1**
Fast Fourier Transform (FFT) - [Instruction 60] **B-2**
File Mark in Storage Module **4-6**, **12-7**
Fill and stop memory **4-7**
Figures, List of **LF-1**
Final Storage
and High/low resolution formats **2-3**
Changing size of **1-4**
Definition **OV-4**, **OV-5**, **2-1**, **A-1**
Erasing **1-5**
Example using two Final Storage areas **8-9**
Format **2-3**, **C-2**
Output data resolution & range limits **2-3**
Ring memory **2-1**
Flags **3-3**
Displaying and toggling flags **1-3**
Output and Program Control **3-3**
With J, K commands **C-1**
Floating point **2-3**, **3-1**
Fractional Value - [Instruction 44] **10-3**
Full Bridge with Excitation Compensation - [Instruction 9] **9-5**
Programming examples **7-7**, **7-13**
Full Bridge with Single Differential Measurement - [Instruction 6] **9-4**
Programming examples **7-10**, **8-11**
Full Duplex, Definition **6-7**
Functional Modes **OV-10**, **1-1**

G

Garbage **6-8**, **A-1**
Geokon's Vibrating Wire Pressure Sensor **7-15**
GraphTerm (Terminal Emulator Program) **OV-9**
Ground loop influence on resistance measurements **13-21**
Grounding
Lightning protection **OV-4**, **14-6**
see also Analog, Earth ground power
Gypsum Soil Moisture block, 227 **7-13**

H

Half Duplex, Definition **6-7**
Handshaking on 9 pin connector
Definition **A-1**
Modem/terminal hardware **6-5**
High resolution
and Memory size **2-1**
Definition **A-1**
Range limits **2-3**
Histogram - [Instruction 75] **11-3**

I

If Case X < F - [Instruction 83] **12-1**
If Flag/Port - [Instruction 91] **12-5**
Programming examples **8-2**, **8-6**
IF Then/Else comparisons **3-4**
If TIME - [Instruction 92] **12-5**
Programming example **OV-17**
If X Compared to F - [Instruction 89] **12-4**
Programming examples **8-3**, **12-4**

If X Compared to Y - [Instruction 88] **12-4**
Increment Input Location - [Instruction 32] **10-1**
Indexed Input Location, Definition **A-1**
Indexing Input Locations and ports **3-1**, **A-1**
Indirect Indexed Move - [Instruction 61] **10-6**
Initiate Telecommunications - [Instruction 97] **12-7**
Input Locations Indexing **3-1**
Input Storage
Altering **1-3**
Changing size of **1-5**
Data format **2-3**, **C-2**
Definition **OV-5**, **A-2**
Input/Output Instructions **9-1**
Definition **OV-7**, **A-2**
Memory and execution times **3-7**
Programming examples **7-1**
Voltage range parameter **3-2**
INT8 Interval Timer, see SDM-INT8
Installation and maintenance **14-1**
Power requirements **14-1**
Instruction location number, Definition **A-2**
Instruction set, CR10 **3-1**
Definition **OV-7**
Format **OV-11**
see Input/Output, Output Processing, Program Control
Integer data type parameter **3-1**
Integer portion, Extracting [Instruction 45] **10-3**
Integer Value - [Instruction 45] **10-3**
Integration time **13-1**
Intermediate Processing Disable Flag **3-4**
Intermediate Storage
Changing size of **1-4**
Data format **2-3**
Definition **OV-5**, **A-2**
Internal Data Storage **2-1**
Internal Memory **OV-5**
Internal Temperature - [Instruction 17] **9-8**
Programming example **OV-12**
Inverse - [Instruction 42] **10-3**
Interval Timer, see SDM-INT8

K

Key functions **OV-9**
Keyboard, See CR10KD Keyboard/Display

L

Label Subroutine - [Instruction 85] **12-1**
Programming examples **8-6**, **8-11**
Program Table **1-1**
Lead acid power supply **14-2**
Leads, See Cables/Leads
Li-Cor LI200S Silicon Pyranometer **7-1**
Library Options Not in Standard Manual **B-2**
Lightning, Protection from **14-6**
Ln(X) - [Instruction 40] **10-2**
Load cell
Programming examples **7-12**, **8-9**
Load Fixed Data - [Instruction 30] **10-1**
Programming example **7-19**
Logarithmic Sampling **8-10**
Loop - [Instruction 87] **12-2**
Counter, Definition **A-2**
Definition **A-2**
Index, see Step Loop index - [Instruction 90]
Programming examples **8-3**, **12-4**
Low Pass Filter - [Instruction 58] **10-6**
Low resolution
and Memory size **2-1**
Definition **A-2**
Range limits **2-3**
LP Filter - [Instruction 58] **10-6**
Lysimeter, weighing **7-11**

M

Maintenance **14-8**
Manually initiated, Definition **A-2**
Manually initiated data transfer (*8 Mode) **4-3**
Maximize - [Instruction 73] **11-3**
Programming example **OV-16**
Measurement Programming Examples **7-1**
Memory and Programming Concepts **OV-5**
Allocation **1-5**
Automatic RAM check on power-up **1-4**
Changing RAM or PROM chips **G-1**
Description of areas **OV-5**
Erasing all **1-5**
Expanding **1-5**, **G-1**
Final Storage **OV-5**, **2-1**
Input Storage **OV-5**, **1-5**
Instruction & execution time **3-5**
Intermediate Storage **OV-5**, **1-5**
Internal **OV-5**
Pointers **2-1**
Program **OV-4**, **3-6**
System **OV-4**
Testing and system status - *B **1-6**
Minimize - [Instruction 74] **11-3**
Programming example **OV-17**
Minus sign (-) & (--), Entering **3-1**
Modes, General overview **OV-10**
*0, Compile/Log Data **1-4**
*1, Program Table 1 **1-1**

*2, Program Table 2 **1-1**
*3, Program Table 3 **1-1**
*5 - Set/Display Clock **1-2**
*6 - Display/Alter Memory and Ports **1-3**
*7 - Display Stored Data on Keyboard/Display **2-3**
*8 Manually initiated Data Output **4-3**
Interrupts during **6-3**
Output device codes for **4-2**
*9 Commands to Storage Module **4-8**
*A Internal Memory Allocation **1-5**
*B Memory Test and System Status **1-6**
*C Security **1-7**
*D, Save/Load Program **1-7**
Errors **3-9**
Modem/terminal **A-2**
Computer requirements **6-5**
Definition **A-2**
Peripherals **6-2**
Modem Enable line on CR10 **4-1**, **6-1**
Peripheral requirements **6-4**
Troubleshooting, Connecting to CR10 **6-7**
Modulo divide - [Instruction 46] **10-3**
Mounting options **14-1**
Move Input Data - [Instruction 31] **10-1**
Programming example **8-8**
Move Signature into Input Location - [Instruction 19] **9-8**
Move Time to Input Location - [Instruction 18] **9-8**
MPTR **2-2**
MSX10/MSX20 Solar Panels **14-5**
Multiply
see X*F - [Instruction 37] **10-2**
see also X*Y - [Instruction 36] **10-2**

N

Natural logarithm - [Instruction 40] **10-2**
Negative numbers **3-1**
Nesting **3-5**
Noise
Common sources **13-1**
Modem **5-2**, **6-4**
Rejection **3-2**
Non-integer portion, Extracting **10-3**
Nonlinear thermistor, Connecting to CR10 **7-14**

O

On-line data transfer **4-1**, **A-2**
Operating details **vi**
OR construction, Logical **3-5**
Output Array
and Output Interval **A-2**
Calculating data points per **4-6**
Definition **A-2**
Setting ID **2-2**
Output device codes
for *8 Mode **4-2**
for Instruction 96 **4-2**
Output Flag
Description **3-3**
Example of setting **OV-13**
Setting to interval < 1 minute **8-5**
Output interval **A-2**, **OV-7**
Output Processing Instructions **11-1**
Definition **OV-7**, **A-2**
Memory and execution times **3-6**
Overrange detection **3-2**
Overrun execution **1-1**

P

Parameter
Data types **3-1**
Definition **A-3**
Parameter Extension - [Instruction 63] **10-7**
Parity, Checking **6-7**
Paroscientific pressure transducer **7-19**
PC201 Tape Read Card **4-5**
PC208 Datalogger Support Software **OV-9**
Period Measurement - [Instruction 27] **9-13**
Programming example **7-18**
Peripherals
Current drain for common **14-1**
Enabling/Addressing **6-2**
General **4-1**
pH meter, Connecting to CR10 **7-2**
Physical description, CR10 **OV-1**
Polynomial - [Instruction 55] **10-5**
Port, CR10 37 pin description **D-1**
Serial Input/Output 9-pin **6-1**
Port Read - [Instruction 25] **9-12**
Port Set - [Instruction 20] **9-8**
Ports, See Control/logic ports
Power ground **OV-3**, **14-6**
Power, Raising to Y - [Instruction 47] **10-3**
Power supplies **14-2**
Connecting directly to Wiring Panel **14-5**
Connecting to CR10 **OV-3**, **OV-4**
PPTR **2-2**
Pressure transducer
Programming examples **7-10**, **7-15**
Print device, Definition **A-3**
PRINT option on-line data transfer **4-1**
Print peripherals **4-2**, **6-2**, **A-3**
Printer
Controlling data transmission to **2-2**, **4-1**

Output formats **4-6**
Save/Load programs (*D Mode) **1-9**
Printer Pointer (PPTR) **2-2**
Processing Instructions **10-1**
Definition **OV-6**, **A-3**
Memory and execution times **3-7**
Program Control Flags **3-3**
Program Control Instructions **12-1**
Command code parameter **3-4**
Definition **OV-4**, **A-3**
Logical constructions **3-4**
Memory and execution times **3-8**
Programming examples **7-1**, **8-1**
Program memory
Definition **OV-5**, **1-5**
Signature **1-6**
Program Tables
Compiling **1-2**
Definition **1-1**, **A-3**
Example of entering program **OV-13**
Exceeding execution interval **1-1**
Execution interval **OV-7**
Instruction location number **A-2**
Interrupt subroutine **12-1**
Programming example **8-6**
Start/stop running **1-1**
Table priority **1-2**
Programming
Entering a program **OV-12**
Entering negative numbers **3-1**
Functional modes **OV-10**
Instruction format **OV-11**
Key definition **OV-10**
Logical constructions **3-4**
Manual control of program execution **1-3**
Maximum program size **1-5**
Overrange detection **3-2**
Overview of Instruction Set **3-1**
Program memory **1-5**
Remote **5-4**
Save/load programs (*D Mode) **1-9**
Sequence **OV-11**
Programming examples **OV-12**
Editing an existing program **OV-15**
Sample program 1 **OV-13**
Sample program 2 **OV-16**
Programming the CR10 **OV-10**
PROM
Changing chips **G-2**
Library options **B-1**
Signature **1-6**, **B-1**
Protection
from the environment **14-1**
Lightning **14-6**
Moisture **14-1**
Reverse polarity **14-1**
Temperature **14-1**
Transient **OV-1**, **14-1**, **14-6**
Psychrometer programming example **12-3**
Pulse Count - [Instruction 3] **9-1**
Programming examples **7-6**, **8-5**, **8-6**
Pulse inputs **OV-3**
PVC insulated conductors, Avoiding **13-10**
Pyranometer, Connecting to CR10 **7-1**

R

Rainfall intensity, Example of programming **8-2**
Rain gauge, Tipping Bucket
Connecting to CR10 **7-6**
Counting switch closure on **8-6**
RAM
Changing chips **G-1**
Check on power-up **1-4**, **G-1**
Installing new chips **G-1**
Installing new PROM **G-1**
RC35 Cassette Recorder
Description **4-4**
Typical current drain **14-1**
Read Ports - [Instruction 25] **9-12**
Record Real Time - [Instruction 77] **11-4**
Programming example **OV-16**
Reference junction **13-13**
Relays, Using digital I/O ports for switching **14-7**
Remote Keyboard State **OV-9**, **5-4**
Repetitions parameter **3-1**
Resetting CR10 **vi**, **1-5**
Resistance measurements
Bridge resistance **13-17**
Requiring AC excitation **13-21**
Resolution **2-3**, **11-5**
Retrieval options, Data storage and **OV-19**
RH (207) **7-4**, **9-6**
Ring interrupts **6-3**
Ring Line (Pin 3) **6-1**
Ring memory **2-1**
Run Time errors **3-9**

S

Sample - [Instruction 70] **11-3**
Programming example **OV-13**
Sample on Maximum or Minimum - [Instruction 79] **11-5**
Sample rate **1-1**, **A-3**
Saturation Vapor Pressure - [Instruction 56] **10-5**
SC12 cable **OV-9**
SC32A RS232 Interface **OV-9**, **6-6**

SC90 Serial Line Monitor **4-7**
SC92/93 for writing to tape, Don't use **4-4**
SC92A/93A **4-4**
Scaling Array with Multiplier & Offset - [Instruction 53] **10-4**
Programming example **8-7**
SDC99 Synchronous Device Interface **6-3**
SDM-A04 4 Channel Analog Output Module - [Instruction 103] **9-15**
Current drain, Typical **14-1**
Programming example **8-7**
SDM-CD16 Control Port Expansion Module - [Instruction 104] **9-15**
Current drain, Typical **14-1**
SDM-INT8 8 Channel Interval Timer - [Instruction 101] **9-14**
Current drain, Typical **14-1**
SDM-SW8A Switch Closure Input Module - [Instruction 102] **9-14**
Current drain, Typical **14-1**
SD's, see Synchronous devices
SDM peripherals **7-24**
Security **1-7**
Send Character - [Instruction 98] **12-8**
Sensors
Connecting to Wiring Panel **OV-3**
Effect of lead length on signal settling time **13-3**
Settling errors for CSI resistive sensors **13-8**
Serial Input/Output
General **6-1**
Pin Description **6-1**
Serial Out - [Instruction 96] **4-1**, **12-6**
Programming example **OV-17**
Set Active Storage Area - [Instruction 80] **11-5**
Programming examples **8-2**, **8-3**, **8-9**
Set Resolution Data Storage Format - [Instruction 78] **11-5**
Setting and Displaying the Clock - *5 Mode **1-2**
Spark gaps on wiring panel terminals **14-6**
Signature
CR10 PROM **1-6**, **B-1**
Definition **A-3**
Generation of **C-4**
Signature - [Instruction 19] **9-8**
Sin(X) - [Instruction 48] **10-3**
Single-ended measurements on analog inputs **OV-3**, **13-1**
Single-ended Volts - [Instruction 1] **9-1**
Programming example **7-1**
SM192/716 Storage Modules, see Storage Modules
Sample on Maximum or Minimum - [Instruction 79] **11-5**
Solar panels, Solarex MSX10/MSX18 **14-5**
Spatial Average - [Instruction 51] **10-4**
Spatial Maximum - [Instruction 49] **10-3**
Spatial Minimum - [Instruction 50] **10-4**
Specifications **OV-20**
SPTR, see Storage Module Pointer
Square Root - [Instruction 39] **10-2**
Standard and Weighted Value Histogram - [Instruction 75] **11-3**
Standard Deviation in Time - [Instruction 82] **11-5**
Step Loop Index - [Instruction 90] **12-4**
Stop Bit **6-7**
Storage and retrieval options, Data **OV-19**
Storage Module Pointer (SPTR) **2-2**
Storage Modules (SM192/716) **4-7**
Addressing with CR10 **4-1**
Commands to (*9 Mode) **4-8**
Current drain, Typical **14-2**
File Mark **4-7**
Manually initiated data output (*8 Mode) **4-7**
Save/load program (*D Mode) **1-9**
Use with Instruction 96 **4-2**
Storage peripherals, External **4-1**
Storage
See Final Storage, input storage, intermediate storage
Store Area - [Instruction 80] **11-5**
Programming example **8-3**
Strip charts
Converting wind direction output to 0-540 **8-8**
Use with SDM-A04 **8-7**
Subroutines
Entering **1-1**
Label Subroutine - [Instruction 85] **12-1**
Port Interrupt Subroutines **12-1**
Programming Example **8-6**
Subtract - [Instruction 35] **10-1**
SW8A Switch Closure Input Module, see SDM-SW8A
Switch closure
Counting on Tipping Bucket Rain Gage **8-6**
Measuring **9-1**
Switch Closure Input Module, see SDM-SW8A
Switching power to a device **14-7**
Synchronous Device Communication **6-4**
Synchronous, Definition **A-3**
System memory **OV-5**, **1-4**
System status (*B Mode) **1-6**

T

Tables, List of **LT-1**
see also Program Tables

Tape Pointer (TPTR) **2-2**
Tape recorder
  Connecting to CR10 **4-4**
  Data format for **4-5**
  Dump data (*8 Mode) **4-3**
  Interrupts during transfer **6-3**
  Manually initiated data transfer (*8 Mode) **4-3**
  On-line data transfer (Instruction 96) **4-1**
TPTR (Tape Pointer) **2-2**
Tapes, Recommended **4-4**
Telecommunication **5-1**
  Automatic setting of baud rate **5-1**
  Automatic time-out from **5-2**
  Commands **5-3**
  Key definitions **OV-10**
  Password **5-4**
  Remote keyboard **OV-9**, **5-4**
  Telecommunication states **5-4**
  with Binary responses **C-1**
Telecommunications Command **OV-10**
Telecommunications (Modem) Pointer (MPTR) **2-2**
Temp (107) - [Instruction 11] **9-5**
Temp, Module - [Instruction 17] **9-8**
Temp RTD - [Instruction 16] **9-7**
Temp TC SE - [Instruction 13] **9-6**
Temp TC DIFF - [Instruction 14] **9-7**
Temperature from Platinum R.T.D. -
  [Instruction 16] **9-7**
  Programming examples **7-7**, **7-8**, **7-10**
Temperature from thermocouples, See
  Thermocouple temperature
Temperature of Module - [Instruction 17] **9-7**
  Programming example **OV-13**
Temperature range, CR10 **14-1**
Terminal emulator **OV-9**
Thermocoule Measurements **13-12**
Thermocouple Temperature Measurement -
  [Instruction 14] **9-7**
  Programming example **7-3**
  Single-Ended Voltage - [Instruction 13] **9-6**
Technique/error analysis **13-13**
  Using an External Reference Junction **7-3**
Three Wire Half Bridge - [Instruction 7] **9-4**
  Programming example **7-8**
Throughput rate, Definition **1-1**, **A-3**
Time - [Instruction 18] **9-8**
  Programming example **8-4**
Time
  Setting/displaying (*5 Mode) **1-2**
  Storing in Final Storage **11-4**
Timer - [Instruction 26] **9-13**
Timer, see SDM-INT8 8 Channel Interval
Tipping Bucket Rain Gage **7-6**, **8-6**
Totalize - [Instruction 72] **11-3**
  Programming example **8-3**
TPTR **2-2**
Tutorial **OV-1**

U

Use of Digital I/O Ports **14-7**
User flags **3-3**
Using the PC208 Terminal Emulator
  (GraphTerm) **OV-9**

V

Vapor Pressure From Wet-/Dry-Bulb
  Temperatures - [Instruction 57] **10-5**
  Programming example **12-3**
Vehicle power supply **14-5**
Vibrating wire, measure sensor, Geokon's **7-15**
Vibrating Wire Measurement -
  [Instruction 28] **9-13**
  Programming example **7-18**
Voltage on analog input, excess **vii**
Voltage measurements
  Differential **7-2**
  Differential/single-ended **OV-3**, **13-1**
  Integration **13-1**
  Ranges/codes and overrange detection **3-2**, **9-1**
Volts (SE) - [Instruction 1] **9-1**
Volts (Diff) - [Instruction 2] **9-1**

W

Watchdog reset **3-8**
WDT-VP - [Instruction 57] **10-5**
Wind Vector - [Instruction 69] **11-1**
  Programming example **8-8**
Wiring Panel
  Connecting to vehicle power supply **14-5**
  Description **OV-1**, **14-5**

X

X * F - [Instruction 37] **10-2**
  Programming example **7-19**
X * Y - [Instruction 36] **10-2**
  Programming example **7-22**
X + F - [Instruction 34] **10-1**
X + Y - [Instruction 33] **10-1**
  Programming example **7-19**
X - Y - [Instruction 35] **10-1**
X / (1-X), see Bridge Transform - [Instruction 59]
X / Y - [Instruction 38] **10-2**
X Mod F - [Instruction 46] **10-3**
X^Y - [Instruction 47] **10-3**

Y

YSI 44032 Thermistor source resistance and
signal levels **13-10**, **13-11**

Z

Z = 1/X - [Instruction 42] **10-2**
Z = ABS(X) - [Instruction 43] **10-3**
Z = EXP(X) - [Instruction 41] **10-2**
Z = F - [Instruction 30] **10-1**
Z = FRAC(X) - [Instruction 44] **10-3**
Z = INT(X) - [Instruction 45] **10-3**
Z = LN(X) - [Instruction 40] **10-2**
Z = SIN(X) - [Instruction 48] **10-3**
Z = X - [Instruction 31] **10-1**
Z = X * F - [Instruction 37] **10-2**
Z = X + F - [Instruction 34] **10-1**
Z = X * Y - [Instruction 36] **10-2**
Z = X + Y - [Instruction 33] **10-1**
Z = X - Y - [Instruction 35] **10-1**
Z = X / Y - [Instruction 38] **10-2**
Z = X MOD F - [Instruction 46] **10-3**
Z = X^Y - [Instruction 47] **10-3**